



PIC16F84-Simulator

Dokumentation

Entwicklung im Rahmen der Vorlesung Systemnahe Programmierung

des Studienganges Informatik

Informationstechnik

an der Dualen Hochschule Baden-Württemberg, Karlsruhe

Ersteller:	Nicolas Mansky und Frederik Rueß
Datum:	13.05.2019
Abgabedatum:	13.05.2019

Inhaltsverzeichnis

1. Grundlagen.....	3
1.1. Arbeitsweise eines Simulators	3
1.2. Vor- und Nachteile eines Simulators.....	3
1.2.1. Vorteile	3
1.2.2. Nachteile	3
1.3. GUI.....	4
2. Realisierung.....	5

1. Grundlagen

1.1. Arbeitsweise eines Simulators

Bei einer Simulation wird versucht, ein reales System modellhaft nachzubilden. An diesem Modell können Versuche durchgeführt werden, um auf deren Auswirkungen am realen System Rückschlüsse ziehen zu können. Der im Rahmen der Vorlesung Signale und Systeme erstellte PIC16F84 erstellte Simulator soll hierbei die Ausführung von vorliegenden Beispielprogrammen simulieren. Die Simulation soll eine graphische Oberfläche besitzen, welche dem Benutzer eine einfache Bedienung ermöglicht.

1.2. Vor- und Nachteile eines Simulators

1.2.1. Vorteile

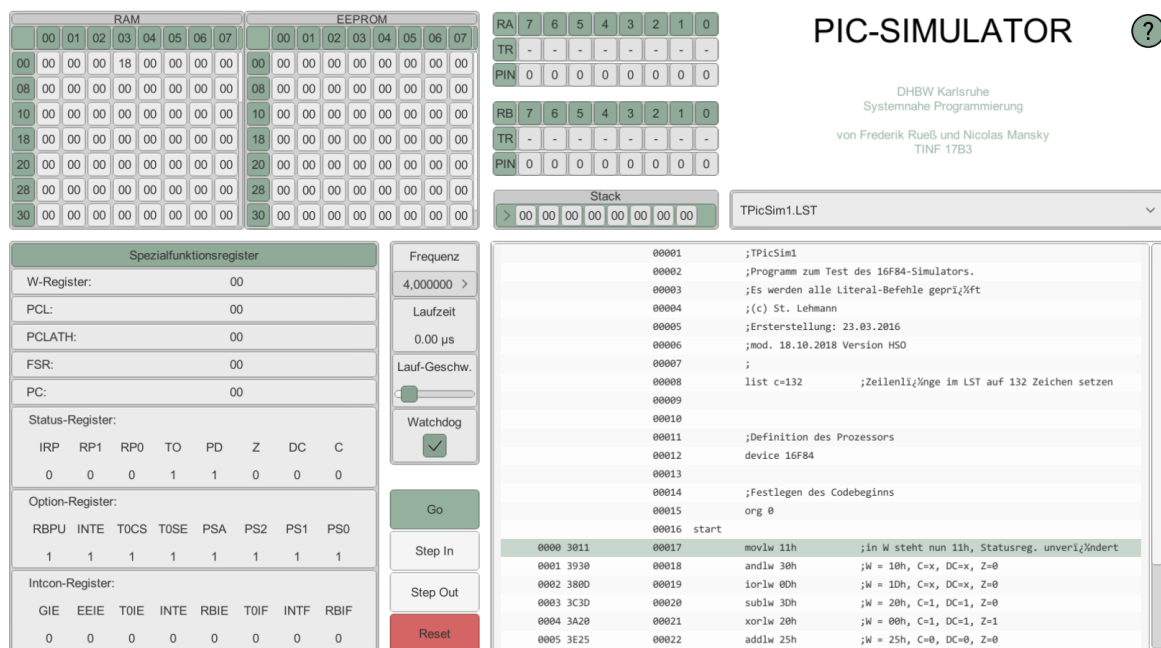
- Modelle können leichter modifiziert werden als reale, bereits bestehende Systeme
- Simulierte Systeme lassen sich besser beobachten
- Der Besitz des nachgebildeten Systems ist nicht notwendig
- Leichte Bedienbarkeit

1.2.2. Nachteile

- Funktionen können ggf. nicht vollständig nachgebildet werden
- Echtzeitsimulation quasi unmöglich

1.3. GUI

Die Benutzeroberfläche des Simulators wurde mit Unity erstellt. Hierbei wurde darauf geachtet, dass das Design sowohl ansprechend als auch für den Benutzer leicht verständlich gestaltet wird.



Die Oberfläche bietet hierbei folgende Funktionen:

- (1) Dropdownmenü zum Einlesen einer LST-Datei
- (2) Viewport zur graphischen Darstellung der eingelesenen Datei
- (3) Start/Stop-Button zum Starten der Simulation
- (4) Step-Button für Schrittweise Ausführung des Codes
- (5) Reset-Button für Soft-Reset
- (6) Quarzfrequenz Einstellen
- (7) Laufzeitmessung
- (8) Laufgeschwindigkeitsregelung
- (9) Watchdog

- (10) Graphische Darstellung RAM-Baustein
- (11) Graphische Darstellung EEPROM-Baustein
- (12) Graphische Darstellung Spezialfunktionsregister
- (13) Graphische Darstellung RA/RB
- (14) Graphische Darstellung Stack
- (15) Hilfe-Button zum öffnen dieser Dokumentation

2. Realisierung

2.1. Programmiersprache

Das Frontend des Simulators wurde mit Hilfe der Unity Engine erstellt. Das Backend wurde mit C# realisiert. C# wurde gewählt, da die Unity Engine am besten mit diesen Skripten arbeiten kann.