

---

ALBERT LUDWIGS UNIVERSITÄT FREIBURG

TECHNISCHE FAKULTÄT

## PicoC-Compiler

### Übersetzung einer Untermenge von C in den Befehlssatz der RETI-CPU

BACHELORARBEIT

*Abgabedatum:* 28<sup>th</sup> April 2022

*Author:*  
Jürgen Mattheis

*Gutachter:*  
Prof. Dr. Scholl

*Betreuung:*  
M.Sc. Seufert

---

Eine Bachelorarbeit am Lehrstuhl für  
Betriebssysteme

---

---

---

## **ERKLÄRUNG**

Hiermit erkläre ich, dass ich diese Abschlussarbeit selbständig verfasst habe, keine anderen als die angegebenen Quellen/Hilfsmittel verwendet habe und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe. Darüber hinaus erkläre ich, dass diese Abschlussarbeit nicht, auch nicht auszugsweise, bereits für eine andere Prüfung angefertigt wurde.

---

---

# Inhaltsverzeichnis

Abbildungsverzeichnis	I
Codeverzeichnis	II
Tabellenverzeichnis	III
Definitionsverzeichnis	IV
Grammatikverzeichnis	V
<b>1 Ergebnisse und Ausblick</b>	<b>1</b>
1.1 Funktionsumfang . . . . .	1
1.1.1 Kommandozeilenoptionen . . . . .	1
1.1.2 Shell-Mode . . . . .	2
1.1.3 Show-Mode . . . . .	2
1.2 Qualitätssicherung . . . . .	3
1.3 Erweiterungsideen . . . . .	3
<b>Literatur</b>	<b>A</b>

---

---

---

# Abbildungsverzeichnis

---

---

# Codeverzeichnis

1.1	Shellaufruf . . . . .	2
-----	-----------------------	---

---

---

# Tabellenverzeichnis

1.1	Kommandozeilenoptionen . . . . .	1
-----	----------------------------------	---

---

---

# Definitionsverzeichnis

---

---

# Grammatikverzeichnis



# 1 Ergebnisse und Ausblick

Zum Schluss soll ein **Überblick** über das gegeben werden, was im Kapitel ?? implementiert wurde. In Unterkapitel 1.1 wird mithilfe **kurzer Anleitungen** ein grober Einblick in die **wichtigsten Funktionalitäten** des implementierten **PicoC-Compilers** und **anderer mitimplementierter Tools** gegeben. Im Unterkapitel 1.2 wird aufgezeigt, was zur **Qualitätssicherung** implementiert wurde, um zu gewährleisten, dass der **PicoC-Compiler** die Kompilierung der **Programmiersprache**  $L_{PicoC}$  in **Syntax** und **Semantik** **identisch** zur entsprechenden **Untermenge** der Programmiersprache  $L_C$  umsetzt. Als allerletztes wird im Unterkapitel 1.3 ein Ausblick gegeben, wie der PicoC-Compiler **erweitert** werden könnte.

## 1.1 Funktionsumfang

Bei der Implementierung des **PicoC-Compilers** wurden verschiedene **Kommandozeilenoptionen** und **Modes** implementiert. Diese werden in den folgenden Kapiteln 1.1.1, 1.1.2 und 1.1.3 mithilfe kleiner **Anleitungen** erklärt werden. Eine ausführliche **Dokumentation** ist unter [Link<sup>1</sup>](#) zu finden.

### 1.1.1 Kommandozeilenoptionen

Will man einfach nur ein **Programm** `program.picoc` kompilieren ist das mit dem **PicoC-Compiler** genauso **unkompliziert** wie mit dem **GCC** durch einfaches **Angeben der Datei**, die kompiliert werden soll: `> picoc_compiler program.picoc`. Als Ergebnis des Kompiliervorgangs wird eine Datei `program.reti` mit dem entsprechenden **RETI-Code** erstellt, wobei für die **Benennung der Datei** einfach nur der **Basisname** der Datei `program` an eine neue **Dateiendung** `.reti` angehängt wird<sup>2</sup>.

Daneben gibt es allerdings auch die Möglichkeit **Kommandozeilenoptionen** `<cli-options>` in der Form `> picoc_compiler <cli-options> program.picoc` mitanzugeben, von denen die wichtigsten in Tabelle 1.1 erklärt sind. Alle weiteren **Kommandozeilenoptionen** können in der **Dokumenation** unter [Link<sup>1</sup>](#) nachgelesen werden.

Kommandozeilenoption	Beschreibung
<code>-i, --intermediate-stages</code>	Zeigt Zwischenschritte der Kompilierung in Form der verschiedenen Abstrakten Syntaxbäume
<code>-p, --print</code>	

Tabelle 1.1: *Kommandozeilenoptionen*

<sup>1</sup>[https://github.com/matthejue/PicoC-Compiler/blob/new\\_architecture/doc/help-page.txt](https://github.com/matthejue/PicoC-Compiler/blob/new_architecture/doc/help-page.txt)

<sup>2</sup>Beim **GCC** wird bei **Nicht-Angabe** eines **Dateinamen** mit der `-o` Option dagegen eine Datei mit der festen Namen `a.out` erstellt.

### 1.1.2 Shell-Mode

Will man z.B. **Folgen von Statements** in der Programmiersprache  $L_{PicoC}$  **schnell** kompilieren ohne eine Datei erstellen zu müssen, so kann der **PicoC-Compiler** im sogenannten **Shell-Mode** aufgerufen werden. Hierzu wird der PicoC-Compiler **ohne Argumente** aufgerufen, wie es in Code 1.1 zu sehen ist.

Mit dem `compile <cli-options> <filename>`-Befehl (oder der Abkürzung `cpl <cli-options> <filename>`) kann **PicoC-Code** zu **RETI-Code** kompiliert werden. Die Kommandozeilenoptionen `<cli-options>` sind dieselben, wie wenn der Compiler **direkt** mit Kommandozeilenoptionen aufgerufen wird. Die wichtigsten Kommandozeilenoptionen sind in Tabelle ?? angegeben.

PicoC Code can be compiled into RETI Code with the 'compile [cli options] <code>;' command (shortcut 'cpl'). The cli options are the same as for calling the compiler from outside, except for the 'infile' argument which is interpreted as string with PicoC Code and which will be compiled as if it was enclosed in a main function.

```
> picoc_compiler
PicoC Shell. Enter `help` (shortcut `?`) to see the manual.
PicoC> cpl "6 * 7;";
----- RETI -----
SUBI SP 1;
LOADI ACC 6;
STOREIN SP ACC 1;
SUBI SP 1;
LOADI ACC 7;
STOREIN SP ACC 1;
LOADIN SP ACC 2;
LOADIN SP IN2 1;
MULT ACC IN2;
STOREIN SP ACC 2;
ADDI SP 1;
LOADIN BAF PC -1;

Compilation successfull

PicoC> quit
```

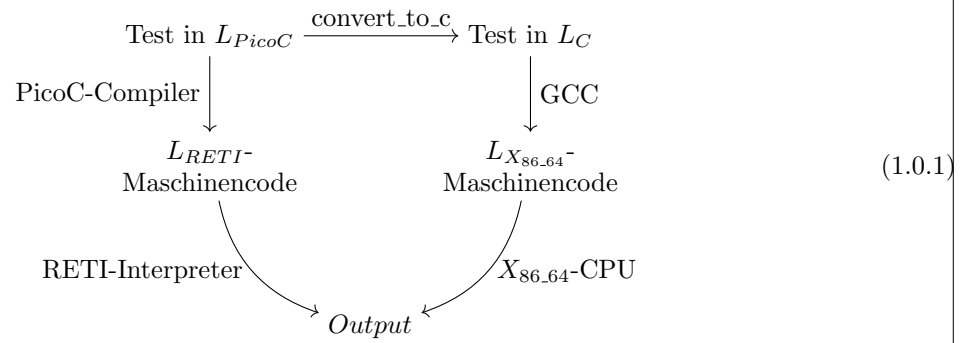
Code 1.1: Shellaufruf

Wenn man möglichst alle nützlichen **Kommandozeilenoptionen** angegeben haben will, bei denen es **keinen** Grund gibt, sie nicht mitanzugeben, kann der Befehl `most\_used` (oder )

### 1.1.3 Show-Mode

Der **Show-Mode** ist ein Nebenprodukt der Implementierung des **PicoC-Compilers**. Dieser **Mode** wurde eigentlich nur implementiert, um beim **Testen** des PicoC-Compilers **Bugs** bei der Generierung des **RETI-Code** zu finden, indem im Terminal eine **virtuell ausgeführte RETI** angezeigt wird, welches den **kompletten Zustand** einer virtuell ausgeführten RETI mit allen **Registern**, **SRAM**, **UART**, **EPROM** und einigen **weiteren Informationen** anzeigt. Allerdings bringt die Möglichkeit des **Show-Mode** die **RETI-Befehle** des übersetzten Programmes in Ausführung zu sehen auch einen großen **Lerneffekt** mit sich, weshalb der **Show-Mode** noch **weiterentwickelt** wurde, sodass auch **Studenten** ihn auf unkomplizierte Weise nutzen können.

## 1.2 Qualitätssicherung



## 1.3 Erweiterungsideen

---

---

# Literatur