

Einrichtung einer C-Entwicklungsumgebung für den Gameboy

Heiko Wolf
heiko.wolf.mail@gmail.com

FDL 1.3

Stand: 03.11.2020

1 Vorwort

Im Jahr 2019 habe ich einen Artikel zur „Einrichtung einer C-Entwicklungsumgebung für den Gameboy“ veröffentlicht¹. Mittlerweile hat sich einiges in dieser Richtung getan, so das ein aktualisierter Text zu diesem Thema an der Reihe ist.

Mittlerweile läuft das GBDK-Toolkit mit einer aktuellen SDCC-Version. Zudem hat GBDK einige Verbesserungen erhalten und auch bei den Tools GBTD und GBMB wurden kleine positive Änderungen vorgenommen.

Im Gegensatz zum ersten Artikel ist nun ein Windows nicht mehr notwendig. Das Vorgestellte läuft unter einer Linuxumgebung, egal ob mit einer Linuxdistribution, der WSL² oder der Linuxumgebung unter ChromeOS³. Unter den drei Genannten habe ich dies getestet. Theoretisch sollte das Ganze auch unter einem BSD⁴ möglich sein.

Für die Realisierung des Vorhabens sind grundlegende Linuxkenntnisse sinnvoll, insbesondere die Arbeit mit der Shell. Programme aus dem Quellcode zu kompilieren sollte keine grundlegende Hürde mehr darstellen. Die Installation und Einrichtung von Wine⁵ sollte bekannt sein.

Als Grundlage wird ein debiantypisches System für diesen Artikel verwendet. Bei Benutzung anderer Systeme werden gegebenenfalls noch zusätzliche Software bzw. Arbeitsschritte benötigt.

Es wird nur auf die Aspekte der Software eingegangen. Die anderen Punkte im Vorgängerartikel gelten nach wie vor.

2 Installation SDCC⁶

Zum Zeitpunkt der Erstellung dieses Dokumentes ist die offizielle Releasenummer von SDCC die 4.0. Mit dieser Version gibt es allerdings einige Probleme mit dem GBDK⁷, so das auf ein Snapshotbuild^{8,9} zurückgegriffen wird.

Wurde der Quellcode eines aktuellen SDCC-Snapshot heruntergeladen und entpackt wird noch zusätzlich folgende Software installiert.

```
sudo apt install g++ libboost-dev texinfo zlib1g-dev bison flex make
```

Danach wird im Verzeichnis des SDCCs der Befehl

```
./configure --disable-pic14-port --disable-pic16-port
```

ausgeführt. Die PIC-Ports benötigen zusätzliche Software. Diese Ports sind aber für die Entwicklung für den Gameboy nicht relevant und werden daher deaktiviert.

¹ <https://drive.google.com/file/d/1-UlgkdVGSoyuQn4Ta7goR6iACdUDD3M/view?usp=sharing>, abgerufen am 02.11.2020

² https://de.wikipedia.org/wiki/Windows-Subsystem_f%C3%BCr_Linux, abgerufen am 02.11.2020

³ <https://chromeos.dev/en/linux>, aufgerufen am 02.11.2020

⁴ https://de.wikipedia.org/wiki/Berkeley_Software_Distribution, abgerufen am 03.11.2020

⁵ <https://www.winehq.org/>, abgerufen am 02.11.2020

⁶ <http://sdcc.sourceforge.net/>, abgerufen am 02.11.2020

⁷ <https://github.com/Zal0/gbdk-2020/blob/develop/README.md> bei Current Status, abgerufen am 02.11.2020

⁸ <http://sdcc.sourceforge.net/snap.php>, abgerufen am 02.11.2020

⁹ https://sourceforge.net/projects/sdcc/files/snapshot_builds/sdcc-src/sdcc-src-20201027-11925.tar.bz2/download, abgerufen am 31.10.2020

Mit dem Befehl

```
make
```

und der Kommandozeile

```
sudo make install
```

wird der Compiler sdcc kompiliert und installiert. Per

```
sdcc -v
```

kann geprüft werden, ob alles in Ordnung ist.

3 GBDK

Das initiale Vorhaben, das Toolkit GBDK auf einer aktuelle SDCC-Version zu bringen, geschah durch Zalo¹⁰ ¹¹. Er ist auch der Entwickler der Gameboy-Spieleengine ZGB¹². Die Änderungen liegen in einem Github-Repositorye.

Mit

```
git clone https://github.com/Zal0/gbdk-2020.git
```

wird der aktuelle Quellcode gezogen.

Wichtig ist es, die Variable SDCCDIR zu setzen, damit das make vom GBDK diesen findet. Sollte bei der Installation von SDCC ein anderer Pfad gesetzt wurden sein, muss dieser entsprechend angepasst werden. Ansonsten reicht

```
SDCCDIR="/usr/local" make
```

um GBDK zu kompilieren und per

```
sudo SDCCDIR="/usr/local" make install
```

zu installieren.

Nun noch das Verzeichnis /opt/gbdk/bin in die PATH-Variable aufnehmen und per

```
lcc -v
```

wird sichtbar, ob alles geklappt hat.

Damit sind die grundlegenden Programme für die Gameboy-Entwicklung eingerichtet.

4 GBTD und GBMB

Der Github-Benutzer Toxa¹³ hat sich den Tools GBTD und GBMB angenommen und die const-Problematik¹⁴ vor den C-Arrays korrigiert und mit Delphi 5 neu kompiliert. Diese sind

¹⁰ <https://zalods.blogspot.com/>, abgerufen am 02.11.2020

¹¹ <https://github.com/Zal0>, abgerufen am 02.11.2020

¹² <https://github.com/Zal0/ZGB>, abgerufen am 02.11.2020

¹³ <https://github.com/untoxa>, abgerufen am 02.11.2020

¹⁴ vor dem generierten C-Arrays der Tile- bzw. Mapdaten fehlte ein const, so das GBDK diese nicht erkannt hatte

auch per Wine¹⁵ lauffähig¹⁶. Die ausführbaren Dateien und weitere Informationen zu den Änderungen sind unter https://github.com/untoxa/GBTD_GBMB/releases zu finden.

5 Nachwort

Mit den neuen Entwicklungen steht die Entwicklung mit C für den Gameboy wieder auf einer soliden Basis. Der Zugriff auf über 20 Jahre alte Software ist nun minimiert und die Abhängigkeit zu Windows reduziert. GBTD und GBMB lassen sich eventuell per Lazarus¹⁷ kompilieren, so dass auch ausführbare Dateien für Linux zur Verfügung stehen und somit Wine wegfallen kann.

¹⁵ getestet mit Wine 3.0.x und 4.0.x

¹⁶ Hilfe lässt sich nach wie vor nicht anzeigen, ist aber eher ein Wineproblem

¹⁷ <https://www.lazarus-ide.org/index.php>, abgerufen am 02.11.2020