

Einrichtung einer C- Entwicklungsumgebung für den Gameboy auf dem Raspberry Pi

Heiko Wolf
heiko.wolf.mail@gmail.com

FDL 1.3

Stand: 03.01.2021

1 Vorwort

Mit dem Raspberry Pi 400 ist der Pi¹ nun völlig in den Gefilde von Desktopumgebungen aufgegangen². In diesem Dokument wird die Einrichtung einer Gameboy-C-Entwicklungsumgebung für den Raspberry Pi betrachtet. Als Referenzmodell steht der Raspberry Pi 4. Theoretisch sollte die Einrichtung auch bei früheren Modellen ab Version 2 funktionieren.

Der Artikel unterscheidet sich von den anderen Einrichtungsartikel³ dadurch, dass unter dem Raspberry Pi kein Wine⁴ für x86 Anwendungen existiert und somit eine zusätzliche Emulationsschicht für die Ausführung von Wine für x86 notwendig ist, um die Windows-Programme GBTD, GBMB⁵ und gegebenenfalls den Gameboy-Emulator BGB⁶ auf dem Raspberry Pi auszuführen.

Für die Realisierung geht dieser Artikel von einem fertig eingerichteter Pi aus. Kenntnisse in den üblichen Linux Befehlen und die Kompilierung von Programmen sollte bekannt sein.

Die verwendete Distribution ist Raspberry Pi OS⁷.

2 Kompilierung von SDCC und GBDK

Für die Entwicklungsumgebung wird die C-Umgebung GBDK⁸ verwendet. Das Originalprodukt ist mächtig veraltet. Mittlerweile gibt es eine aktuellere Entwicklung⁹ die auch für den Raspberry Pi funktioniert. GBDK verwendet als Grundcompiler den SDCC¹⁰. Für die neuere GBDK-Umgebung wird auf ein Snapshot build¹¹ zurückgegriffen, da GBDK mit der aktuellen¹² Releaseversion 4 GBDK Probleme¹³ macht.

Wurde der Quellcode eines aktuellen Snapshots heruntergeladen und das Archiv entpackt. Wird im SDCC-Verzeichnis folgendes ausgeführt:

```
sudo apt install g++ libboost-dev texinfo zlib1g-dev bison flex make
./configure --disable-pic14-port -disable-pic16-port
make
sudo make install
```

Die Kompilierung von SDCC dauert einige Zeit (nicht nur eine Tasse). Ist das Ganze aber abgeschlossen, kann per

```
sdcc -v
```

das Ergebnis angesehen werden.

1 https://de.wikipedia.org/wiki/Raspberry_Pi, abgerufen am 03.01.2021

2 <https://www.raspberrypi.org/products/raspberry-pi-400/>, abgerufen am 29.12.2020

3 <https://drive.google.com/file/d/1T-BhFdF6jTos8HrZApfBmmtQzFoIV-vO/view?usp=sharing>, abgerufen am 03.01.2020

4 <https://www.winehq.org/>, abgerufen am 03.01.2021

5 https://github.com/untoxa/GBTD_GBMB, abgerufen am 03.01.2021

6 <http://bgb.bircd.org/>, abgerufen am 03.01.2021

7 <https://www.raspberrypi.org/software/>, abgerufen am 03.01.2021

8 <http://gbdk.sourceforge.net/>, abgerufen am 29.12.2020

9 <https://github.com/Zal0/gbdk-2020/>, abgerufen am 29.12.2020

10 <http://sdcc.sourceforge.net/>, abgerufen am 29.12.2020

11 <http://sdcc.sourceforge.net/snap.php>, abgerufen am 29.12.2020

12 zum Zeitpunkt der Erstellung dieses Artikels

13 <https://github.com/Zal0/gbdk-2020/blob/develop/README.md>, abgerufen am 29.12.2020

Nachdem SDCC eingerichtet ist wird GBDK kompiliert. Über

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

wird der aktuelle Programmcode bezogen.

Folgende Schritte sind zur Einrichtung von GBDK notwendig (im Verzeichnis vom GBDK)

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

und mit

```
lcc -v
```

kann der Erfolg überprüft werden.

3 Box86

Wie bereit erwähnt kann das mit Raspberry Pi OS gelieferte Wine keine x86-Programme ausführen¹⁴, daher wird zur Ausführung einer x86 Wineversion die Anwendung Box86¹⁵ verwendet. Box86 ist ein x86 Emulator für diverse Arm-Systeme.

Für die Kompilierung¹⁶ ist cmake u.a. notwendig.

Mit

```
git clone https://github.com/ptitSeb/box86  
cd box86  
mkdir build; cd build; cmake .. -DRPI4=1 -DCMAKE_BUILD_TYPE=RelWithDebInfo  
make -j4  
sudo make install  
sudo systemctl restart systemd-binfmt
```

wird Box86, wie in der Dokumentation beschrieben, kompiliert¹⁷ und installiert. Mit dem Befehl

```
box86 -h
```

kann das Ergebnis überprüft werden.

4 Wine

Box86 verwendet viele vorhandene Systembibliotheken. Wine muss daher nur noch heruntergeladen und kann dann mit Box86 ausgeführt werden. Das Projekt TwisterOS¹⁸ stellt ein Tar-Archiv¹⁹ zur Verfügung. Das Archiv einfach entpacken und dann per

```
box86 ~/wine/bin/wine winecfg20
```

14 sondern nur Arm-Programme

15 <https://github.com/ptitSeb/box86>, abgerufen ab 03.01.2021

16 <https://github.com/ptitSeb/box86/blob/master/COMPILE.md>, abgerufen ab 03.01.2021

17 gegebenenfalls müssen noch weitere Programme, wie z. B. gcc, vorher installiert werden

18 <https://twisteros.com/index.html>, abgerufen am 03.01.2021

19 <https://twisteros.com/wine.tgz>, abgerufen am 03.01.2021

20 Pfad entsprechend anpassen

ausführen.

Die Tools GBMB und GBTD können nun mit

```
box86 ~/wine/bin/wine gbmb.exe
```

bzw.

```
box86 ~/wine/bin/wine gbtd.exe
```

ausgeführt werden. Die Performance ist dabei völlig in Ordnung.

TwisterOS²¹ bringt Box86 gleich mit, bei Verwendung dieser Distribution kann das Kompilieren von Box86 und Wine hinzufügen wegfallen.

5 Gameboy Emulator

Als Emulator könnte BGB in Verbindung mit Box86 und Wine verwendet werden. Der Start des Programms dauert aber ewig und der Author hat bisher kein Programmfenster zu Gesicht bekommen²². Es kann aber auch ein nativer Emulator wie z. B. mgba²³ benutzt werden. Die in Raspberry Pi OS mitgelieferte Version von mgba ist veraltet. Eine aktuelle Version lässt sich per Flatpak²⁴ installieren. Über

```
sudo apt install flatpak
wget https://dl.flathub.org/repo/appstream/io.mgba.mGBA.flatpakref
flatpak install io.mgba.mGBA.flatpakref
```

wird der Emulator auf dem Raspberry Pi installiert.

Mit

```
flatpak run io.mgba.mGBA.flatpakref25
```

wird mgba ausgeführt.

Mit Sameboy²⁶ existiert als Beispiel noch ein weiterer aktueller Emulator. Diese Anwendung müsste über den Quellcode kompiliert werden. Aktuell bietet er unter Linux-Systemen aber keine Funktionen zum Debuggen an. Wirbt allerdings mit einer guten Genauigkeit²⁷.

6 Schlußwort

Mit den in diesem Artikel aufgezeigten Schritten ist es nun auch möglich auf dem Raspberry Pi eine C-Entwicklungsumgebung für die Gameboy-Plattform einzurichten und zu verwenden.

21 dem Author war diese Distribution allerdings zu überladen

22 kann aber an einer eingestellten Programmoption liegen

23 <http://mgba.io/>, abgerufen am 03.01.2021

24 <https://dl.flathub.org/repo/appstream/io.mgba.mGBA.flatpakref>, abgerufen am 03.01.2021

25 Befehlszeile kann auch ins Menü integriert werden

26 <https://sameboy.github.io/>, abgerufen am 03.01.2021

27 <https://sameboy.github.io/features/>, abgerufen am 03.01.2021