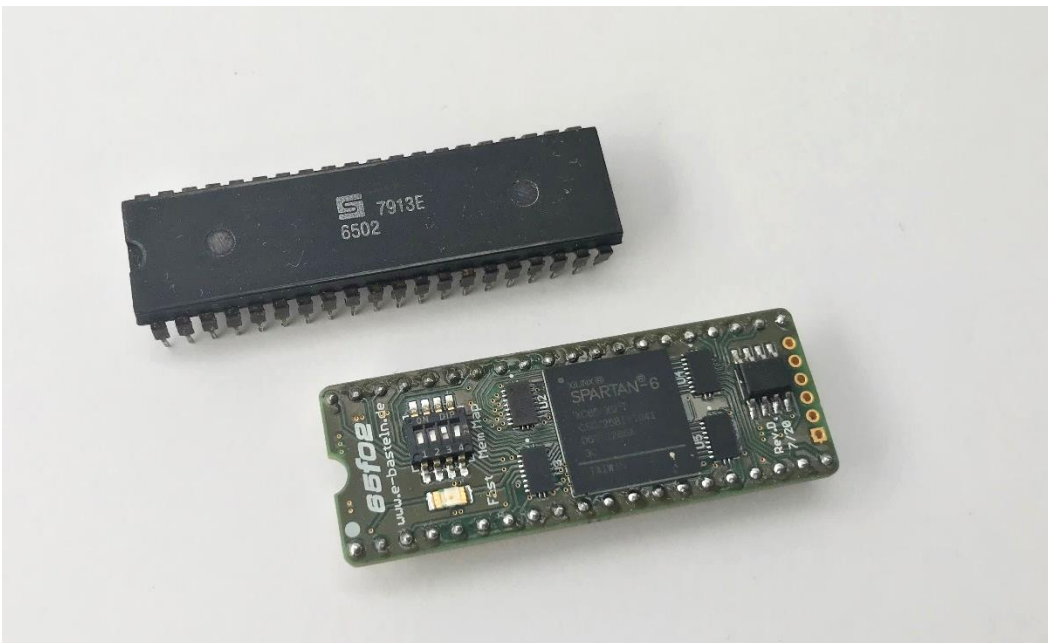


# Einbau- und Bedienungsanleitung 65F02 Rev D

Roland Langfeld, Jürgen Müller

20.8.2020, 22.11.2020

65F02 ist eine FPGA-basierte Prozessoremulation, die Pin-kompatibel zum 65C02 ist, intern aber mit 100 MHz läuft. Diese Anleitung gilt für die Version Rev D.



Kontakt:

Roland Langfeld

[R.Langfeld@t-online.de](mailto:R.Langfeld@t-online.de)

Jürgen Müller

[juergen@e-basteln.de](mailto:juergen@e-basteln.de)

# Inhalt

Einbau .....	3
Voraussetzungen.....	3
Platine einsetzen .....	4
Einstellung der DIP-Schalter .....	5
Platine ausbauen.....	6
Optional: Schalter für Originalgeschwindigkeit .....	6
Benutzung.....	8
Normaler Betrieb .....	8
Fehlersuche.....	8
Funktion der LED .....	9
Updates des FPGA.....	10
TinyProg-Software installieren .....	10
USB-Adapter aufstecken und anschließen.....	10
TinyProg-Software nutzen.....	11
Lizenzen und Rechtliches .....	13
Open-source Komponenten .....	13

# Einbau

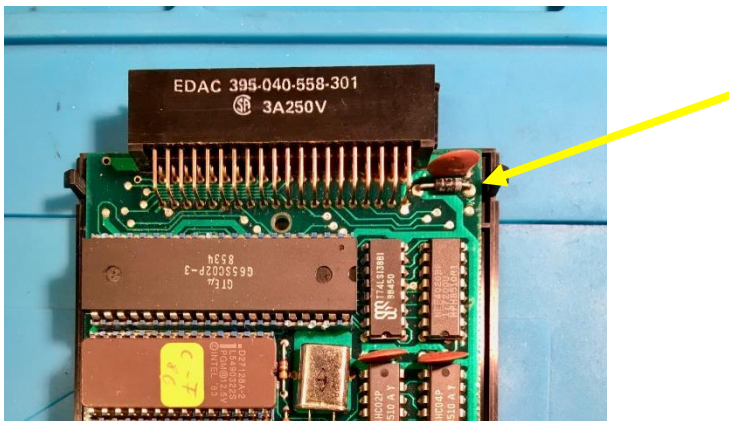
## Voraussetzungen

Die 65F02 kann derzeit folgende Schachcomputer auf 100 MHz beschleunigen:

- Mephisto Modular II & B&P
- Mephisto Rebell 5.0
- Mephisto Modular IV & V, Rebel Portoroz (MM X)
- Mephisto Polgar
- Mephisto Milano & Nigel Short
- Novag Constellation Forte A/B
- Novag Super Constellation
- Chafitz ARB & MGS

Die CPU des Zielsystems muss gesockelt sein. Die 65F02 wird einfach anstelle der Original-CPU in den Sockel eingesetzt; weitere Anschlüsse sind nicht erforderlich. Der Stromverbrauch der 65F02 liegt um 20 bis 30 mA höher als bei der Original-CPU; das können alle Schachcomputer problemlos liefern.

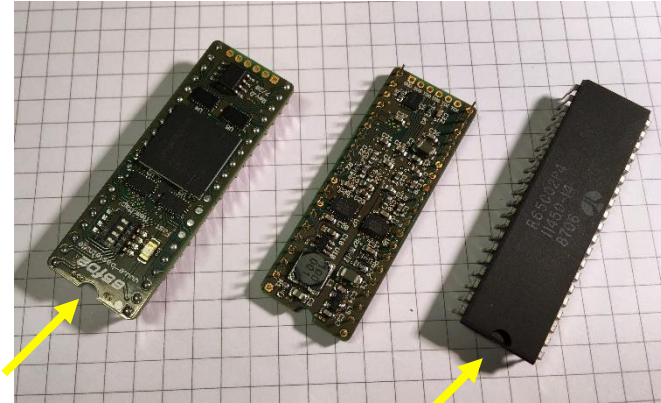
Im Mephisto Modul-System mit der älteren 6 V Brettelektronik kann die 65F02 eingesetzt werden, wenn die Original-Schutzdiode im Modul vorhanden ist (s. Bild 1)



**Bild 1:** Lage der Schutzdiode auf dem MM II (B&P und Rebell 5.0 ähnlich)

## Platine einsetzen

- Gerät vom Netz trennen
- Orientierung der 65C02-CPU merken (halbrunde Kerbe an einer Schmalseite, s. Bild 2). Falls das mal vergessen wurde: Die Markierung findet sich in der Regel auch am CPU-Sockel und als Bestückungsaufdruck auf der Hauptplatine.



**Bild 2:** Orientierungsmarken

- Auf statische Entladung achten: Arbeitsplatz mit einem Stück Alufolie als Arbeitsunterlage bedecken, sich hinsetzen, Alufolie berühren
- Schachcomputer öffnen
- Original-CPU am besten mit schmalen Schraubenzieher o.ä. wechselseitig an den Schmalseiten stückweise hochhebeln und vorsichtig abnehmen
- CPU auf leitfähigem Kunststoffschäum oder einem Stück Alufolie ablegen
- 65F02-Platine vorsichtig erstmal lose auf die leere Fassung legen
- Sitzen alle Pins sauber über den Gegenstücken der Fassung?
- Jetzt vorsichtig am besten mit 3 Fingern ganzflächig auf den Chip drücken und diesen gleichmäßig in die Fassung drücken (s. Bild 3)
- Nochmal genau prüfen, ob alle Pins an der richtigen Stelle in der Fassung sind.



**Bild 3:** gleichmäßig auf den Chip drücken zum Einsetzen

## Einstellung der DIP-Schalter

DIP-Schalter nur in stromlosem Zustand ändern, Änderungen werden beim Einschalten gelesen. Die kleinen Schiebeschalter rasten deutlich ein; sie lassen sich z.B. mit einem Zahnstocher oder Uhrmacherschraubendreher gut betätigen.

Achtung: Getestet sind bisher nur die Einstellungen für alle Mephisto-Typen sowie für den Novag Constellation Forte A/B!

Position

1234

0000 für alle Systeme: 65F02 läuft mit Systemtakt, KEINE Beschleunigung

1000 Mephisto Modular II & B&P

0100 Mephisto Modular IV & V, Rebel Portoroz (MM X)

1100 Mephisto Polgar

0010 Mephisto Milano & Nigel Short

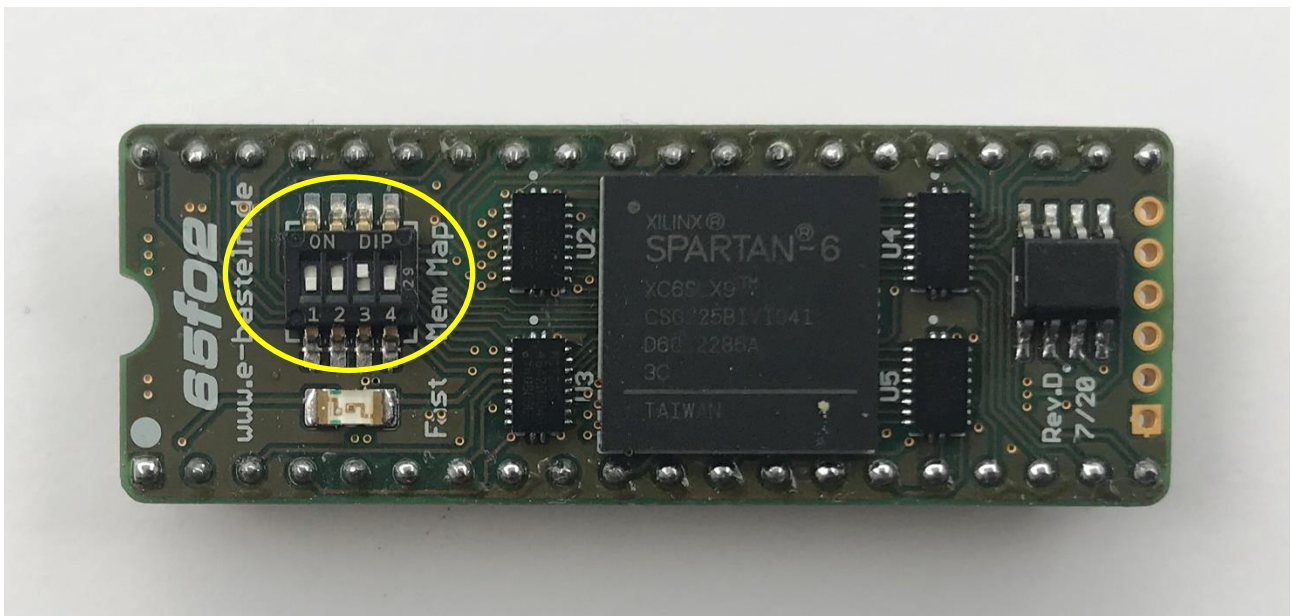
1010 Mephisto Rebell 5.0

0110 Novag Constellation Forte A/B

1101 Novag Super Constellation

1110 Chafitz ARB & MGS

0101 Fidelity Elite A/S (Glasgow, Budapest); Prestige – in Arbeit



Lage der DIP-Schalter. Schalterstellung "1" = "ON" (oben).

## Platine ausbauen

Sinngemäß umgekehrt wie der Einbau, aber ACHTUNG:

Auch auf der Unterseite der 65F02-Platine befinden sich Bauteile.

Daher muss die Platine wie folgt aus der Fassung gehebelt werden:

- NIEMALS an den Schmalseiten hebeln oder ansetzen, sondern
- mit einem breiten Schraubenzieher an den Längsseiten zwischen IC-Fassung und überstehender 65F02-Platine ansetzen,
- ganz wenig durch Drehen der Klinge die Platine ca 1 mm anheben,
- jetzt mit dem Schraubenzieher reihum an den beiden Längsseiten mm für mm die Platine gleichmäßig raushebeln.
- Das braucht mehrere „Umläufe“. NICHT an einer Stelle gleich ganz raushebeln und den Chip verkantet hochhebeln!!



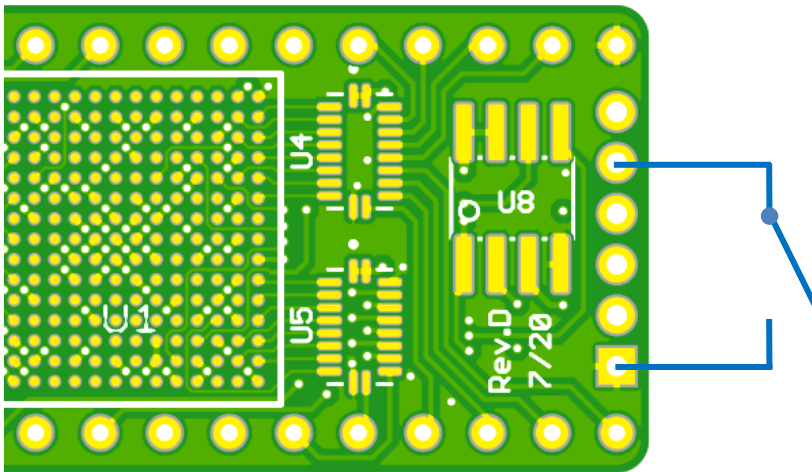
*Bild 4 a,b: Reihum an den Längsseiten vorne und hinten Stück für Stück die Platine hochhebeln*

- Alte CPU mit richtiger Orientierung wieder einsetzen.  
Auch hier auf verbogene Pins achten
- Modul schließen, einbauen, testen, fertig!

## Optional: Schalter für Originalgeschwindigkeit

Zu Test- oder Vergleichszwecken kann es wünschenswert sein, die 65F02 zwischen Originalgeschwindigkeit und beschleunigtem Betrieb umzuschalten. Dazu kann optional ein Schalter angeschlossen werden zwischen den Pins 1 (GND) und 5 (TDI) der Kontaktleiste an der Schmalseite der 65F02. Die Lage der Anschlüsse ist unten gezeigt; auf der Unterseite der Platine sind auch die Pin-Namen zu finden.





- Ist dieser Schalter geschlossen, dann werden die DIP-Schalter „überstimmt“. Die 65F02 läuft dann ohne Beschleunigung, genau wie bei DIP-Schalterstellung „0000“.
- Wie die DIP-Schalter sollte auch dieser Schalter nicht während des Betriebs betätigt werden. Das führt zu undefinierten Betriebszuständen (beschädigt aber nichts).
- Ein einfacher Schließer genügt, z.B. auch ein Reedkontakt.
- Anschlussdrähte können entweder direkt an die Lötungen gelötet werden oder über eine Steckerleiste. Achtung: Die Pins haben den etwas ungewöhnlichen Abstand von 2,0 mm! Gerade oder gewinkelte Stiftleisten in diesem Rastermaß sind verfügbar, aber der gängige Standard von 2,54 mm passt nicht. Auch Stecker/Buchse-Kombinationen sind erhältlich, z.B. vom Hersteller JST.
- Wird die Umschaltung der Geschwindigkeit nicht benötigt, dann kann die Kontaktleiste einfach ungenutzt (offen) bleiben.

# Benutzung

## Normaler Betrieb

- Modul im Brett wie gewohnt einschalten:
- erst das Netzteil in die Steckdose
- dann Schachcomputer am Brett einschalten
- Gerät ist wie bisher zu bedienen
- Ausschalten: erst am Brett, dann Netzteil aus der Steckdose

Einige Schachcomputer haben batterie-gepuffertes RAM, um eine Partie nach dem Aus- und Einschalten des Geräts wieder aufnehmen zu können. Dies wird von der 65F02 bisher nicht unterstützt, da sie nur das interne, schnelle RAM im FPGA verwendet. Voraussichtlich können wir diese Funktion durch ein künftiges Update des FPGA-Codes nachrüsten, aber derzeit gilt:

- Partien gehen beim Ausschalten stets verloren.
- Evtl. kann beim Einschalten des Geräts auch ein Reset bzw. „New Game“ erforderlich sein, um einen definierten Ausgangszustand herzustellen.

## Fehlersuche

In den Mephisto-Schachcomputern ist die 65F02 gründlich getestet; einige der anderen Modelle sind frisch implementiert und konnten nur oberflächlich oder gar nicht getestet werden. Bei Problemen gern bei Jürgen melden!

Zur ersten Fehlersuche folgende Tipps:

- Stimmt die DIP-Einstellung??
- Probeweise auf 0000 stellen – läuft es dann?
- Bei offenem Modul schauen: leuchtet die LED auf der Platine?  
Details zur LED siehe unten!
- Sind einzelne Pins neben die Fassung gerutscht?
- Ist der Chip tief genug in die Fassung gesteckt?
- Ist der Chip richtig herum eingebaut?  
(runde Platinenkerbe ist da, wo auch die CPU-Kerbe war)
- Falls immer noch kein Ergebnis:  
Rückbau auf orig. 65C02 – läuft es dann?



## Funktion der LED

Die LED zeigt an, wie effektiv die 65F02 den Zielrechner gerade beschleunigt. Sie kann auch bei der Eingrenzung von Fehlern helfen:

- Beim Start (Strom ein) blinkt die LED stets einmal kurz auf. Sollte sie stattdessen komplett dunkel bleiben, erhält die 65F02 wahrscheinlich keine Versorgungsspannung.
- Anschließend leuchtet die LED um so heller, je stärker die CPU beschleunigt wird. Folgende Anzeigezustände sind normal:
  - LED aus – keine Beschleunigung (normal bei Schalterstellung 0000)
  - LED hell – schnelles Rechnen mit 100 MHz
  - Mittlere Helligkeit oder Flackern – schnelles Rechnen und intensive Zugriffe auf die Original-Peripherie (z.B. Display, Lautsprecher) wechseln sich ab.
- Gleichmäßiges Pulsieren der LED mit ca. 1 Hz zeigt an, dass das FPGA bereit für ein Update via USB ist.
  - Dieser Zustand ist normal, wenn der USB-Programmieradapter aufgesteckt wurde (s. unten).
  - Tritt das Pulsieren beim Betrieb im Schachcomputer auf, dann signalisiert das, dass das Reset-Signal des Schachcomputers nach dem Einschalten ständig anliegt. Das deutet auf einen Einbaufehler hin.

# Updates des FPGA

Neue Konfigurationen für das FPGA können nachträglich per USB-Anschluss eingespielt werden. Damit können bei Bedarf Fehler behoben oder neue Funktionen bereitgestellt werden.

- Ein USB-Adapter liegt der 65F02 bei.
- Benötigt wird zusätzlich ein USB-Kabel (Steckertyp A am Computer, Steckertyp Mini-B an der 65F02) und
- die frei erhältliche Programmiersoftware „TinyProg“, die unter Windows, Mac OS oder Linux lauffähig ist.

## TinyProg-Software installieren

Die 65F02 nutzt die Open-Source-Software TinyProg von Luke Valenty (<https://github.com/tinyfpga/TinyFPGA-Bootloader/tree/master/programmer>).

TinyProg ist in Python geschrieben. Das hat den Vorteil, dass das Programm unter allen gängigen Betriebssystemen verwendet werden kann; es erfordert aber die Installation der Python-Programmierungsumgebung.

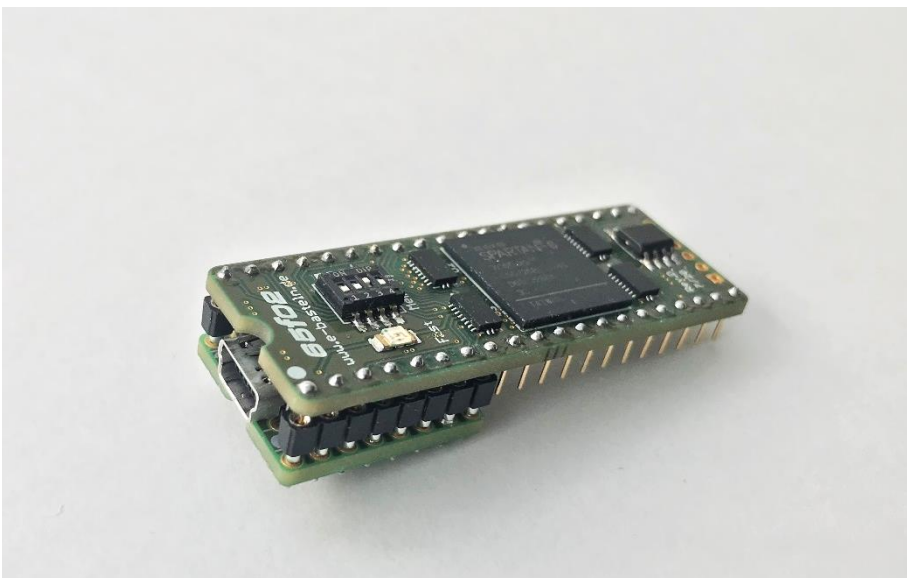
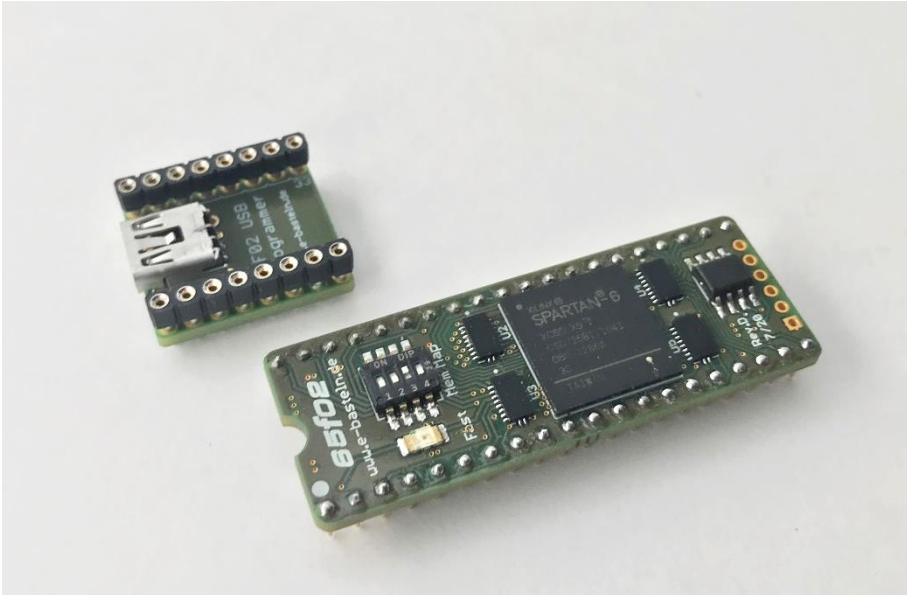
- Aktuelle Python-Version installieren von <https://www.python.org/>. Getestet wurde mit Python 3.8.3.
- Kommandozeile bzw. Shell aufrufen
- TinyProg installieren mit `pip install tinyprog`. (Unter MacOS stattdessen `pip3.8 install tinyprog`.)
- Testen, ob TinyProg richtig installiert wurde: `tinyprog -h` gibt Kommandoübersicht aus.
- Unter MacOS muss noch zusätzlich eine fehlende Bibliothek sowie ein Paketmanager installiert werden:
  - HomeBrew Paketmanager installieren von <https://brew.sh/>
  - USB-Bibliothek installieren mit `brew install libusb`

## USB-Adapter aufstecken und anschließen

- Die 65F02 aus dem Schachcomputer ausbauen wie oben beschrieben.
- Der USB-Adapter wird auf der Seite der 65F02 aufgesteckt, die mit der Kerbe markiert ist. Die Bilder unten zeigen die richtige Position.
- Anschließend USB-Kabel (Type Mini B) an den Computer anschließen.

Zum späteren Abziehen des USB-Adapters entsprechend vorgehen wie beim Ausbau der 65F02 aus dem Computer:

- USB-Verbindung trennen
- Adapter mit einem Schraubenzieher vorsichtig an den Längsseiten abhebeln, nicht verkanten.



## TinyProg-Software nutzen

- USB-Kabel anschließen. Die 65F02 wird via USB mit Strom versorgt. Die LED beginnt regelmäßig zu pulsieren.
- Kommandozeile bzw. Shell des Betriebssystems öffnen

- `tinyprog -m`  
gibt Information zum angeschlossenen FPGA-Modul aus.  
Es sollten „boardmeta“ und „bootmeta“-Daten angezeigt werden.
- `tinyprog -p dateiname.bin`  
programmiert das FPGA neu mit der angegebenen Binärdatei.  
Das Binärformat ist spezifisch für das Spartan-6 FPGA.  
Valide Binärdateien sind stets 340 604 Bytes lang!

Das Programmieren sollte wenige Sekunden dauern; Status und Fortschritt werden angezeigt. Nach dem Programmieren einfach das USB-Kabel abziehen, den USB-Adapter trennen wie oben beschrieben, und die 65F02 wieder in den Rechner einbauen.

Sollte die Programmierung ausnahmsweise fehlschlagen, so bleibt der Boot-Loader im FPGA trotzdem erhalten. USB-Verbindung für einige Sekunden trennen, wieder anstecken und erneut versuchen.

## Lizenzen und Rechtliches

Die Urheberrechte für diese Anleitung, das 65F02 Platinendesign und die Software (mit Ausnahme der unten genannten Open-Source-Komponenten) liegen bei

© 2020 Jürgen Müller und Roland Langfeld.

Die kommerzielle Nutzung erfordert unsere schriftliche Genehmigung. Die nicht-kommerzielle Nutzung wird hiermit gestattet; wir würden uns freuen, über die Erfahrungen mit der 65F02 zu hören.

Die 65F02 ist ein Hobbyprojekt und kein kommerzielles Produkt. Wir geben bei Problemen mit dem Nachbau oder der Anwendung gern Hilfestellung, können aber keine formale Garantie oder Produkthaftung übernehmen.

### Open-source Komponenten

Die **Verilog 65C02 CPU**, die im 65F02-FPGA eingesetzt wird, wurde von Arlet Ottens als 6502-Kern entwickelt und von Ed Spittles und David Banks um die 65C02 Erweiterungen ergänzt. Der Quelltext für Arlets ursprüngliche 6502 ist auf [Arlets Webseite](#) verfügbar, die erweiterte 65C02-Version auf [github](#), und technische Diskussionen dazu im [6502.org Forum](#).

Vielen Dank an Arlet, Ed und David, die ihre Arbeit als Open Source Software unter den folgenden großzügigen Lizenzbedingungen freigegeben haben:

(C) Arlet Ottens, (C) 2016 David Banks and Ed Spittles  
Feel free to use this code in any project (commercial or not), as long as you keep this message, and the copyright notice. This code is provided "as is", without any warranties of any kind.

Der **TinyFPGA USB Boot-Loader** und die dazugehörigen TinyProg Programmiersoftware stammen von Luke Valenty.

Der [TinyFPGA Boot-Loader](#) ist auf der 65F02 vorinstalliert und wird unter der [Apache License Version 2.0](#) weitergegeben. Das [TinyProg](#) Python-Programm, das zur FPGA-Programmierung auf dem PC gestartet wird, kann aus seinem offiziellen Online-Archiv installiert werden wie im vorhergehenden Abschnitt beschrieben.

Vielen Dank an Luke für die Entwicklung dieser eleganten Programmierlösung und für die Veröffentlichung als Open Source Software!