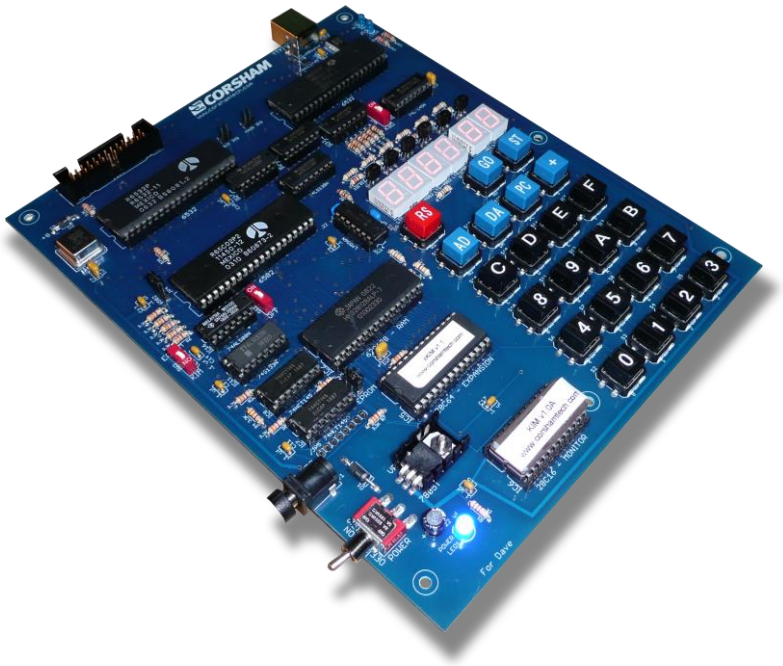
A PROGRAMMER'S GUIDE TO KIM PROGRAMMING

EINLEITUNG



Einleitung

Es stellt sich an dieser Stelle sicherlich den meisten die Frage, warum sich jemand mit einem Thema beschäftigen sollte, das so weit in der Vergangenheit liegt und heutzutage absolut sinnlos scheint. Im Zeitalter der schnellen Computer, wo das Kaufen eines Rechners beim Erreichen der Kasse den Umstand in sich birgt, dass das Gerät zu diesem Zeitpunkt garantiert schon einen Nachfolger hat und technisch gesehen überholt ist. Und dennoch habe ich mich dafür entschieden - vielleicht ist auch ein bisschen Nostalgie im Spiel - Zeit zu investieren, einer tot geglaubten Technik neues Leben einzuhauchen. Auf dieser Bahn bin ich scheinbar nicht der Einzige, denn die Retro-Szene boomt und findet täglich mehr Anhänger. Wer sich heute mit der Programmierung von Mikrocontrollern befassen will, der hat es zum Beispiel mit dem Arduino sehr einfach und kann darüber einen geeigneten Einstieg finden. Dort kann in der Hochsprache C/C++ programmiert werden und alles ist fein. Doch immer weniger Programmierer wissen, was eigentlich im Hintergrund passiert und was der Compiler so mit dem Quellcode anstellt. Wer macht sich eigentlich noch Gedanken über Dinge wie Adressbus, Datenbus, Steuerbus, RAM, ROM und wie diese Komponenten zusammenwirken? Natürlich gibt es diese Themen auch heute noch, doch sie werden vor uns im Verborgenen ihren Dienst tun und fast niemand kennt die genauen Zusammenhänge. Warum auch?!

In diesem Guide möchte ich mich der Programmierung der 6502 CPU widmen und nutze dazu den *KIM-1*. Der KIM-1 ist ein auf dieser CPU basierender Einplatinenrechner, der vom US-amerikanischen Hersteller Commodore International ab den 1976er Jahren vertrieben wurde. KIM steht für *Keyboard Input Monitor*. Das System besitzt einen 2 KB großen Festwertspeicher (ROM), in dem das Betriebssystem untergebracht ist und einen Arbeitsspeicher von 64 KB (RAM) mit diversen Ein- bzw. Ausgabeports. Zu Programmieren ist dieser Rechner über eine Eingabeeinheit mit 24 Tasten und einer sechsstelligen Siebensegmentanzeige.

Die Ein- bzw. Ausgabeports können dafür genutzt werden, um diverse Peripheriegeräte wie z.B. Drucker, Terminal oder einen Kassettenrecorder zur Speicherung der Programme bzw. Daten anzuschließen.

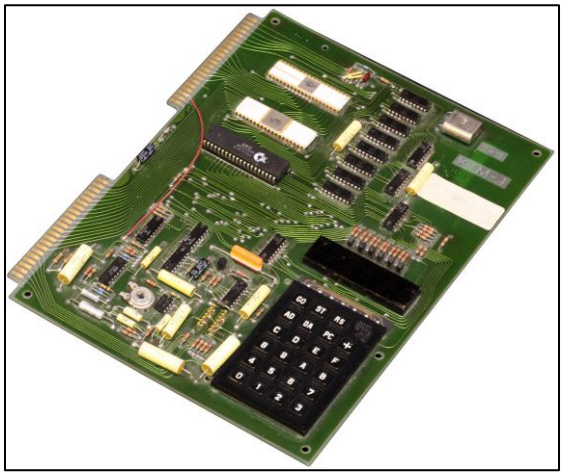


Abbildung 1: KIM-1 (Quelle: Wikimedia)

Im Internet finden sich zahlreiche - teilweise auch Original - Unterlagen, wie z.B.

- KIM-1 User Manual
- The first Book of KIM (FBOK)
- KIM hints

Ein entsprechender Suchbegriff in Google liefert zahlreiche Treffer.

Zur Programmierung einer 6502 CPU gibt es natürlich vielerlei Möglichkeiten und Ansätze und es ist dabei nicht unbedingt erforderlich, sich die passende Hardware zu besorgen. Apple II bzw. C64 Emulatoren gibt es wie Sand am Meer und darüber sind natürlich ohne Probleme die ersten Schritte in dieser Richtung möglich, denn diese Computer nutzen intern den 6502 bzw. 6510. Für den Apple II gibt es den *AppleWin*-Emulator und für das Programmieren in Maschinensprache eignet sich der Merlin Macro-Assembler von Glen Bredon wohl sicherlich am besten.

Abbildung 2: Der Merlin Marco-Assembler#

https://en.wikipedia.org/wiki/Merlin_(assembler)

Ich möchte jedoch einen anderen Weg beschreiten und auf realer Hardware die ersten Schritte zur Programmierung des 6502 durchführen. Dazu ist der KIM-1 hervorragend geeignet. Höre ich da die ersten Rufe: „Wo bekomme ich denn ein derart altes Teil überhaupt her?“ Das ist ein wohl berechtigter Einwand. Es gibt dazu natürlich mehrere Lösungen. Ein Blick in die elektronische Bucht - auch eBay genannt - fördert hier und da vereinzelt, wenn auch manchmal sicherlich überteuert, etwas zutage. Zwei weitere Ansätze sind möglich. Es gibt einen KIM-Clone mit Namen *KIM Uno*, der über einen Arduino (Mikrocontrollerboard) realisiert wurde und entweder bestellt oder auch recht schnell selbst gebaut werden kann. Die notwendigen Informationen sind unter der folgenden Internetadresse zu finden:

http://obsolescence.wixsite.com/obsolescence/kim-uno-summary-c1uuh

Ich habe mir diesen Computer an einem Samstagnachmittag mit wenigen Teilen selbst zusammengebaut, wie das auf der folgenden Abbildung zu sehen ist:

Abbildung 3: Der KIM Uno

Die Folientastatur ist relativ schnell z.B. über Word erstellt und mit einem Laminiergerät in Folie geschweißt. Abschließend noch ein Gehäuse drum herum und fertig ist der KIM-1 Computer, mit dem wunderbar programmiert werden kann.

Der zweite Ansatz ist der Nachbau eines KIM-1 von Bob Applegate, der schon 1976 mit dem KIM-1 seine ersten Berührungspunkte mit Computern hatte. Nähere Informationen sind auf seiner Internetseite unter der folgenden Adresse zu finden:

http://www.corshamtech.com/

Natürlich habe ich mir diesen Clone sofort bei Bob bestellt und werde mit ihm die zahlreichen Beispiele zur Programmierung des 6502 absolvieren. Sicherlich ist es kein Problem, die Beispiele ebenfalls auf dem KIM Uno ans Laufen zu bringen. Das hängt vom jeweiligen Geldbeutel ab, denn der KIM Uno ist für ein paar Euro selbst herzstellen, wohingegen der Kim-1 Clone mit ca. 255 Dollar, das sind ca. 220 Euro, zu Buche schlägt. Es kann sein, dass der Zoll erbarmungslos zuschlägt. Also aufpassen!

Auf der folgenden Abbildung ist der KIM-1 Clone zu sehen und er kommt im Aussehen dem Original schon recht nahe.

Abbildung 4: Der KIM-1 Clone von Bob

Natürlich werden wir uns den Computer noch aus der Nähe anschauen, denn es gibt schon ein paar Unterschiede zum Original KIM-1, die nicht verschwiegen werden sollten.

Wenn es um die Erstellung von Maschinensprache geht, dann ist ein Assembler bzw. Cross-Assembler sicherlich eine sehr große Hilfe. Ich nutze für meine Beispiel den *Ophis Assembler* für die 65xx Serie. Informationen dazu gibt's unter der Internetadresse:

https://michaelcmartin.github.io/Ophis/

Folgende Vorgehensweise schlage ich vor. Ich werde in diesem Guide für die Maschinensprache des 6502 nicht erst die Grundlagen vorstellen und dann später mit den Beispielen aufwarten, sondern wir werden mehr oder weniger direkt einsteigen und ich liefere alle erforderlichen *just-in-time*. Das hat dann nicht den Lehrbuchcharakter, wie man das sonst aus der Schule gewohnt ist und macht in meinen Augen mehr Sinn.

Aber um ein bisschen Grundlagen werden wir wohl nicht herumkommen ☺

Viel Spaß dabei...

Erik Bartmann

http://erik-bartmann.de/