





Ok, ich bekenne mich schuldig: Es hat mich wieder gepackt und ich bastle momentan an einem neuen Softwaresystem für den Hive. Das System wird eine Alternative zum TriOS und besitzt als Kern eine eigene Programmiersprache, die ich ganz unerotisch auf den Namen **m** getauft habe. Im Gegensatz zu TriOS, Spin und PropForth ist es ein völlig neuer Ansatz, dessen Ziele man in vier Worten zusammenfassen kann: autark, klein, schnell und einfach.

Autark: Das System soll sowohl Interpreter, Compiler, Debugger, Quelltexteditor enthalten und mit diesen Funktionen soll es als Betriebssystem TriOS komplett ersetzen, oder unter TriOS als normales Programm gestartet werden können.

Klein: Ziel ist es, den wesentlich größeren Teil des Speichers in Regnatix für die eigentliche Anwendung zur Verfügung zu stellen.

Schnell: Die Kernroutinen sind in PASM geschrieben. Ein erster Vergleich ergab, dass **m** deutlich schneller als Spin und PropForth ist.

Einfach: Ein kleiner und übersichtlicher, klar verständlicher Befehlssatz.

Diese Kernziele geben der Sache ein Bild, aber sie sagen noch nicht alles über den Hintergrund der Sache. Schon die Hardware des Hives ist zumindest in Aspekten neben dem reinen Spaß an der Freude ein medienarchäologisches Experiment, um eine Brücke zwischen Retro- und Neocomputern zu schlagen. Auf der einen Seite stehen dabei Dinge wie Einfachheit und Beherrschbarkeit, auf der anderen Seite Parallelisierung, moderne Speichermedien und Vernetzung. Die Hardware bietet das Potential, die Software ist auf dem Weg.

TriOS war dabei die erste folgerichtige Entwicklungsstufe der Software für mich. Ziel war es, ein gänzlich auf der Sprache Spin basierendes System zu schaffen. Spin ist eine sehr einfache, objektorientierte Sprache und sehr gut geeignet, um in die Technik der gewählten Mikrocontroller einzusteigen. Diese Einfachheit von Spin an der Oberfläche wird erkauft durch Komplexität im Verborgenen: so sind die Spin-Compiler sehr komplex und damit letztlich an einen Host gebunden.

Mit **mental** nun habe ich ein System vor Augen, welches autark arbeitet: Es übernimmt Aufgaben eines einfachen Betriebssystems, enthält eine Scriptsprache, einen Interpreter und Compiler. Mit einem integrierten Quelltexteditor und Funktionen zur Verwaltung der Quelltexte auf dem Datenträger kann ab einer gewissen Entwicklungsstufe die Entwicklung autark auf dem Hive fortgesetzt werden.

Das System kann komplett aus den ROM's gestartet werden und es sollen wie auf den Retros einfache Befehle zur Verfügung stehen, um interaktiv mit einfacher Grafik, Sound und Netzwerk zu experimentieren. Forthähnliche Sprachen sind dafür ganz besonders geeignet, da sie all diese Eigenschaften vereinen und dabei auch noch sehr ressourcenschonend sind.

Mit PropForth gab es in diese Richtung einen ersten Versuch, aber da diese Implementierung von Forth darauf ausgelegt ist, auf nur einem Mikrocontroller zu laufen, nutzt es die Besonderen Ressourcen des hive in keiner Weise. So ist es auch nicht verwunderlich, dass es letztlich bis auf die Autarkie in alle drei anderen Punkten weit hinter meinen Vorstellungen zurück bleiben muß. Das ist nicht als Kritik zu verstehen, da PropForth ja mit ganz anderen Zielsetzungen entwickelt wurde und diese auch sehr gut erfüllt. Nach wie vor ist aber PropForth in TriOS integriert und kann in diesem System mit installiert und getestet werden.

Was also bleibt, ist eine völlige Neuentwicklung. So der here Plan. Wir werden sehen, in welche Richtung sich die Sache entwickelt.

Dokumentation: Um ehrlich zu sein, habe ich keine große Lust Dokumentationen zu schreiben, da ich viel lieber an **mental** schraube. Nun ja, ohne Dokumentation wird aber nichts. Also habe ich mir ein System erweiterbarer Texte erdacht, welche quasi mit dem System wachsen, gleichzeitig aber auch als Tutorial dienen und zusätzlich im Blog als Beitrag erscheinen können.

Die Texte sind chronologisch geordnet, enthalten in ihrer Fußzeile immer einen Hinweis, auf welche Version von **mental** sie sich beziehen und stellen so auch eine Art Logbuch der Entwicklung dar. Es ist also möglich, das ein älterer Text nicht mehr dem aktuellen System entspricht.

Hier gebe ich nur eine kurze Übersicht, der mit dem Erscheinen neuer Texte erweitert wird:

1 - Der Weltraum, unendliche Weite

beschreibt die Entstehung des Kernsystems von **mental** und geht dabei auch auf Implementierungsdetails ein. Es handelt sich quasi um einen Besuch im Maschinenraum des Raumschiffs "**mental**" auf seinem weiten Weg zu neuen Abenteuern...

2 - Der erste Außeneinsatz

ist quasi ein "erste-Schritte-Text" für die Alphaversion. Beschrieben werden die Installationsarten, Hinweise zum Umgang mit dem Dateisystem von **mental** und ein paar erste Fragmente über den Umgang mit dem System an Beispielen.

X - Referenz

Eine umfangreiche Tabelle mit den aktuellen Referenzen zu Befehlen, Steuercodes usw. Dieser Text wird immer aktuell auf die neuste Version bezogen sein.