

---

Subject: Now That I have my CP/M 68K board up and running, What Do I Do Now?  
Posted by [computerdoc](#) on Sat, 25 Apr 2020 03:19:53 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hi CP/M 68K Enthusiasts!

Back in the day when I 1st heard of CP/M, though it interested me, I never was able to put a CP/M system together for the 8080 much less Z80 or 68K.

So my question is: Now that I have my Tiny68K and MB012 boards up and running, what do I do with it?

I know this group has a lot of CP/M users here.

My Background:

I have a strong hardware background with Motorola 8-bit microprocessors. I cut my microprocessor teeth on an MC6800 based ET-3400 Trainer back in college in the '79-'81 Technical College school year which I started when I was 19 years old. Man! I wish I knew then what

I know now about microprocessor electronics and building SBCs! We hand assembled our programs in class back then, entered them on a hex key pad into ram and executed them. That was the development cycle I had to work with. We had no permanent storage or printer of any kind. So remembering those programs was left to taking notes and those are probably long gone now.

Needless to say, our programs in class were hand written in MC6800 Assembly Language and just very basic programs they were. No high level languages for sure, not even a BASIC Interpreter on there.

I don't remember ever getting much training in assembly language never mind training in various basic or advanced concepts & algorithms and we definitely did not learn any type of Disk Operating System because the college either didn't have the advanced Disk drive add-on or it had not been created yet for the ET-3400 though later on years later I heard a 6809 upgrade to the ET-3400 had been made as well as a mini disk drive unit. Microprocessor hardware and software training was quite basic back then - in SC anyway.

I had met a gentleman years before that when I was a kid who is a very dear friend of mine now that had Motorola Technical Databooks so I would read the microprocessor databook in my spare time over at his house for years passionately interested in building a 6800 single board computer eventually though I did not know what that was back then. I almost built a 6800 wirewrapped board with a DB25 RS-232C interface for a terminal with cassette I/O and a parallel port running the Mikbug monitor program for my final project, but it never got finished due to my low confidence level in my Microprocessor Electronics Knowledge and brand new wirewrapping techniques.

I don't know if I still have that board anymore, but I still would have liked to have finished that 6800 board and connect it up to 1 of 2 ASR-33 teletypes with paper tape reader and punch that I used to have. Paper tape was my storage system for a BASIC programming class in that same college though that is a story for another time. Anyway, I still would like to build an MC6802 or MC68HC11 sbc running Flex-02 or Flex-11 if there is such a thing. The 68HC11 has an extra index register if memory serves over the single index register version in the 6800/2/8 chips. I have been collecting many variations of the 68HC11 for years for just such a purpose!

After I found out about the 6809 in the Radio Shack Color Computer 1 (Coco 1) I had bought on sale, I had also wanted to eventually build an MC6809 system running Flex09, but the SWTPC computer systems were priced way out of my earning abilities. Thanks to an individual in this group I was able to procure 5 PCBs to make a 6809/6309 based Uniflex system which as I understand it is a multiuser / multitasking operating system like the OS-9 Disk Operating system I had running on my Coco system. Having these 5 PCBs to build a 6809 UNIFLEX system is a dream come true! Many thanks to that individual!

Many PCBs would be built and much practice creating 6809 based SBCs in Eagle Pro would take place bringing me up to the present day so no CP/M experience of any kind was never had.

Now my question is: How do I learn CP/M 68K style single user version on a Tiny68K or MB012 board SBC? All ideas, suggestions, software projects, etcetera would be most welcome from all of my microprocessor electronics buddies on this forum would be most welcome. Please be nice as I am mostly a hardware guy. I have some programming experience in the past in 8 different programming languages including 6800 assembly language, 8080 assembly language, very little C, C++ Compiler, Expect, Linux batch files, IBM JCL, RPG II, Dbase II, and of course several BASIC Interpreters over the years as well as ADA in the Air Force and a classified programming language for the computer on-board the AWACS E-3 aircraft.

So I have seen and used a lot, but again no CP/M of any kind ... until now.

How do I start programming in C on the Tiny68K and MB012 SBCs?

What software projects should I look into?

What C programming tutorials or training is available these days?

And most important:

What software is available for CP/M 68K that will run in a 16MBs of ram environment on a SBC system? I bet you all wish you had all that ram way back then? 8) :o :roll:

Any takers on these questions?

There are a lot of board builders here realizing now what they wished they could have had back then or did have back in the day and there are others who are building SBCs for the very 1st time.

I think some sort of CP/M 68K and C training should be setup for us newbies to CP/M 68K.

What is the best way to learn proper C programming these days? I did get a copy of The Original C Programming Language book a few years back. I remember seeing it in the college library once or twice. Read about every issue of Byte magazine I could get my hands on, well much of the ads anyway back in the day when BYTE Magazine was HUGE!!! I loved reading the ads scoping to see what was available in the way of 6800 and 6809 computer systems!

Well, if you have read this far, you are at least as interested in the hardware and/or software aspects of and for microprocessor based SBCs as I am or more so.

Where Do I start?

What do you think?

Please, serious thoughts, suggests, ideas, etcetera are warmly welcomed! Take care my friends!

Kip

Computerdoc

Subject: Re: Now That I have my CP/M 68K board up and running, What Do I Do Now?

Posted by [kman](#) on Sun, 26 Apr 2020 09:37:10 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hi,

I think this question applies to most other SBC projects. It's the general question all hobbyist's are facing after finishing a project and have to answer at least. The question is really, where is the threat you have to pick up to get into business with new projects.

When I build my Z80 system in 2018 I realize very early that the terminal output of CPM isn't very exciting. From the 1980's I remember that I install a nice pixel graphic board on my video genie that was really handy. But there where no usable software for it. This thought gave me today a good reason to start the graphic project on my Z80 FPGA system and next the needed software library... At present I'm working on porting a HPGL interpreter for my system. It's a challenging task and the project fills two gaps:

missing graphic software and as a side effect completing my graphic library with some still missing functions.

The Tiny68k is in the same situation, a (may be) boring terminal output on your PC or Laptop and as Kip Koon said, missing software... We have a very nice VT100 terminal board here in our forum. I do not use it, because my Multincomp has Grant's minimal build-in terminal but why not use this and add more functionality to it's firmware (if possible, I do not know the terminal in deep) or program software that uses the existing features increasing the usability of both: terminal & SBC.

CPM3 has this nice RSX feature, which let's one add in a very easy way missing drivers and other software peaces that are missing/needed. Does CPM68k offer something similar ?

I recently get some of these small sensor boards with I2c interface. The idea is to build with a couple of them a minimalistic weather station incl. software: All with the usual things for temperature, moisture, pressure, magnetic field ... I can supply a ANSI-C library for software I2c bit-banging, if needed. Some porting to get it running with your preferred C-Compiler may be necessary (I know, I can buy somethong similar on AMAZON, by why go people fishing....?)

I would change the question from "what can I do with my vintage system", to "there is a long list of projects \_for\_ my system and too less time for all that to do".

A possible task list, others may have/define a completely different one ;-)

cheers  
kurt

---

---

Subject: Re: Now That I have my CP/M 68K board up and running, What Do I Do Now?

Posted by [computerdoc](#) on Fri, 01 May 2020 14:26:54 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hi Kurt!

Thank you for sharing! I appreciate it very much!

Kip

---

---

Subject: Re: Now That I have my CP/M 68K board up and running, What Do I Do Now?

Posted by [computerdoc](#) on Fri, 01 May 2020 15:41:07 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hi Everyone!

I have had 1 wonderful reply from Kurt, "Thank you Kurt!" and 153 views of this topic, "Thank you to everyone who has read this topic thus far!" I get the feeling this is a very important aspect of building Single Board Computers that at least 153 builders seem to be interested in. In light of this interest, I would like to restate my initial request a different way in the hope that many people will share their experiences running the CP/M 68K Operating system as well as other Operating Systems for the 68K line of microprocessors. I imagine there is a lot of great information builders have come across shared by others, created by themselves, etcetera that would be beneficial to all CP/M 68K users in particular and CP/M users in general as well as other 68K operating systems that 68K builders may be interested in. I know of at least 5 Oses that have been written for the 68K series of MPUs over the years. Some are more applicable to 68K based SBCs than others, but 5 - 68K Oses nonetheless! In no particular order, other than memory recall, they are CP/M 68K, Fuzix, OS-9 68K, Commodore Amiga Oses for several of the 68K series of processors and the Macintosh 68K series of Operating Systems. I'm sure there are other 68K Operating systems somewhere in internet land as well. This is a topic where everyone shares in a kind manner how they adapted CP/M 68K, Fuzix and/or OS-9 68K as well as may be other Oses to run on their 68K SBC including successes and those things that did not quite work correctly. A ginormous amount of wonderful information potentially could be shared that just may inspire more people to get involved in learning about microprocessors by building their own SBCs. What do you think?

I would like to hear from everyone of you! I enjoy reading microprocessor manuals anyway so I think I will begin reading the CP/M 68K manuals and see what happens! Take care my friends!

Kip

---

---

Subject: Re: Now That I have my CP/M 68K board up and running, What Do I Do Now?

Posted by [coredump](#) on Fri, 08 May 2020 19:44:40 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hi computerdoc, hi forum,

computerdoc wrote on Sat, 25 April 2020 05:19Hi CP/M 68K Enthusiasts!

Now my question is: How do I learn CP/M 68K style single user version on a Tiny68K or MB012 board SBC?

the CP/M user's guide should be able to guide You.

And in the foreword on page iv it tells what can be found in the other manuals.

You will have to know how to get a directory listing,

for what the user numbers can be (mis)used to,

erasing files,

copy files with pip,

starting programs and submit files.

Using stat, especially to check for file system consistency.

And using the editor (microemacs). Don't use 'ed' from the CP/M distribution. It's only fun for teletype users and those with a somewhat special sense of humor.

And, most important, do not waste Your time by 'learning' it.

Just print out a quick reference for microemacs from the net, make a few notes related to the other commands and have the manuals at hand.

You will have it in memory after a short time by just using it.

Quote:

So I have seen and used a lot, but again no CP/M of any kind ... until now.

How do I start programming in C on the Tiny68K and MB012 SBCs?

What software projects should I look into?

What C programming tutorials or training is available these days?

With some knowledge of the basics, You will most probably be very comfortable with Kernighan & Ritchie 'The C Programming Language' that You already have.

But be sure it is the first edition. The second edition won't work together with the Alcyon C compiler!

Just in case, here is the link:

[https://\\_\\_archive.org\\_details\\_TheCProgrammingLanguageFirstEdition\\_mode/2up](https://__archive.org_details_TheCProgrammingLanguageFirstEdition_mode/2up) (replace underlines with slashes)

But I would like to recommend first tinkering with assembly language.

At times You will need to have a look what the compiler generates (for example to find somewhat hard to find mistakes in Your programs).

And the debugger DDT 'speaks' assembler to You.

Start with useless but funny assembler programs that generate some console output to become familiar with the tools (assembler, linker, debugger).

Find out, how to generate a CP/M OS.

Modify the BIOS.

Start with a new welcome message.

Implement buffers for the UART to switch to an interrupt driven communication.

Make them large (several KBytes), You have plenty of RAM.

Try to implement it fast, but always keep it simple and understandable.

Now you can execute a stop command in the BIOS in the conin or conout routine when the appropriate buffer is empty/full and the CPU would busy wait.

Find out, how the current consumption changes (a stopped CPU does not access the DRAM).

The RTS handshake from UART to USB adapter is no longer urgently required with irq and buffer.

So You can try to switch it to 115.200 Baud to get a faster up/download:

Have the DUART Datasheet at hand, find out it's register addresses.

You need the address for Command Register A (CRA).

Disconnect the RTS physically, hook up an oscilloscope on the RTS output of the DUART.

Read the Command Register A (DDT?).

Note the frequency (oscilloscope), coming now out of the RTS pin.

Set the DUART to 1200 Baud.

Note the increase in frequency at the RTS pin. It should increase from ~0,6 MHz to ~1,8 MHz.

Set the USB adapter to 115.200 Baud.

Implement it in the BIOS.

Having gained some experience have a look at the execution times of the 68000-commands.

Use this when toying with the 'dma' transfer code in the BIOS. You will have fun finding simple ways to make this even faster.

The buffer is always at the same address. And this is usually an even one to be able to access the CF card 16Bit.

With an odd address to transfer to/from, you need to do it byte by byte.

With an also even address, you can do it longword wise.

You can copy down to top with

```
move.l (a0)+,(A1)+
```

or top down with

```
move.l -(a0),-(a1)
```

or free some registers and use

```
movem.l
```

Having four or eight move-commands in the loop, looping only 1/4 or 1/8 times save something on

the time required for branching on the 68000.  
But inhibits loop mode on the 68010/68012.

Just find out what is fastest.

Implement a clever but simple buffering scheme.

Enhance the MB012, build a simple floppy controller card.  
Maybe with FIFO (like the 74ACT2235, but a cheaper one might be better ;- ) to avoid disabling the interrupts.

Implement the driver in the BIOS.  
Enhance your buffering scheme.

You still have a free slot in MB012.  
Maybe a simple ethernet board. Get Geoff tiny\_ftp from the March 1994 Walnut Creek CD (netsrcs/tiny\_ftp) and enhance on that. It's so simple, it can do something even under CP/M.

Or a crt-keyboard card to get rid of the PC with USB-thing as terminal? Have look at the HD63484 ACRTC that is a cool partner of every 16BIT 68000.

Or a new variant of the tiny-boards with the SCC68070? A microcontroller with Philips idea of an 68010 core.

Offering two DMA-channels to speed up CF-access and a simple MMU to hinder self written applications rubbing out the OS or modified disk buffers.  
Or to welcome Minix that would need to be adapted to the hardware an the MMU.

You will have so many ideas yourself that You can share after you have started tinkering with the hardware You already have. Just Do it!

Best regards  
Detlef

---

Subject: Re: Now That I have my CP/M 68K board up and running, What Do I Do Now?

Posted by [Yves-D.](#) on Mon, 11 May 2020 08:48:06 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

I had the same thoughts about missing software for my 68k SBC as ComputerDoc.  
In my OS-search, I skipped CPM-68k as there are virtually no applications available. Looking at the prompt after boot isn't very satisfying in the long run.  
OS-9 68k piqued my interest but seems to be very obscure. Besides some commercial offerings for niche solutions (traffic control, embedded realtime), there's no application software either and no free sources.  
Fuzix shows recent development activity for the core OS, gives access to source code, but is still

in very early beta and again...no applications.

It looks like your bound to:

- a) porting EHBASIC/68k to your SBC
- b) porting Linux/68k to your SBC. There have been some documented portings in this forum to Plasmio's 68030 hardware, which is quite a daunting task and needs in-depth knowledge of Linux and its core-drivers.

I doubt that there will ever be such a gem like RomWBW(Z80) for 68k-SBC's  
Yves