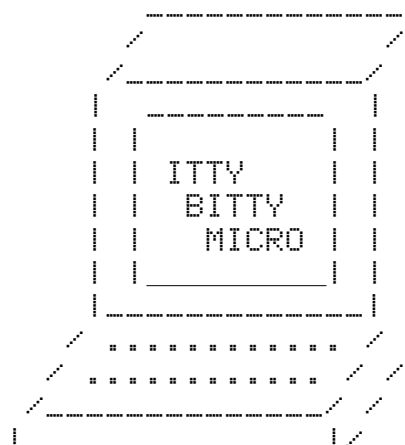


Itty Bitty Micro Company
8080 Future Drive
Rochester, NY 14650



July 10, 1981

Dear Chip,

Here's my first draft of the Assembly manual for our new Z80-SIO card. I tried to include all your notes, but am sure I missed some things (I don't understand half the stuff you do). But point 'em out, and I'll fix them in the next revision.

Sorry about the breathless prose. But ya know, sizzle sells! I tried to include some babes, but Tracey wouldn't put up with it.

Crash is still working on the software. The new ROM is so cool! It boots CP/M in SECONDS instead of minutes like those Microsoft boxes.

But we need more CP/M software for the SD-card. A computer without software is like a car without gas. Osborne and Kaypro are peddling their boxes with tons of free programs! We need to compete.

Anyway, this should work for a "beta test" of the complete kit. Check it out, have someone put it together according to the instructions, and get it working. If all is well, we go to production!

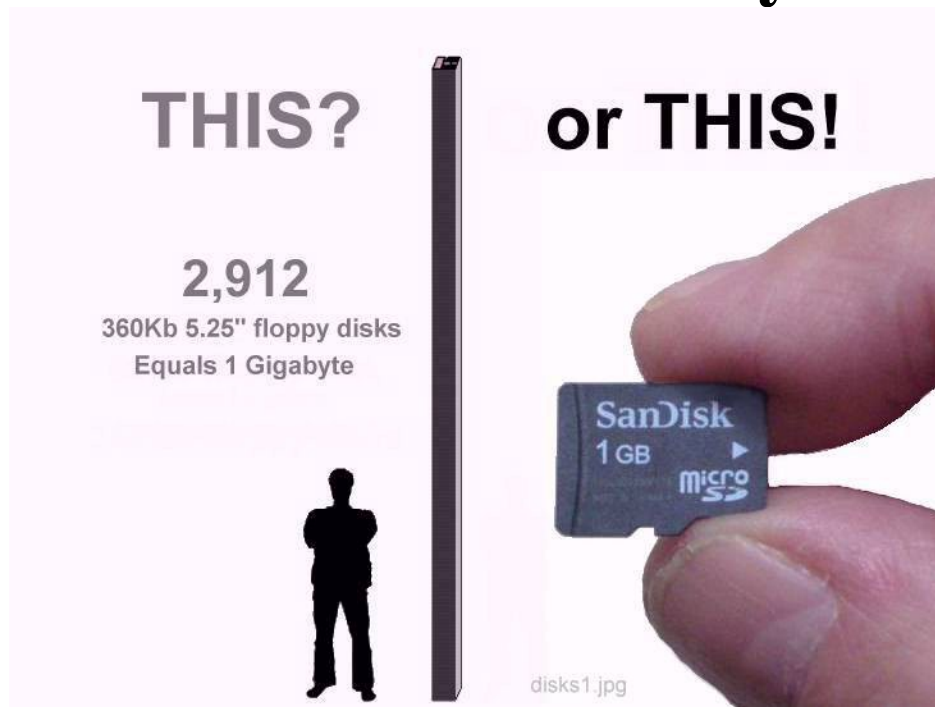
I STILL don't know where the hell you got these memory cards. You said they just use the naked memory chips, without a DIP package and all the pins and stuff that's normally used to put them on a PC board. But still, what have you got: a time machine?

Yours truly,

Gil Bates
Emperor of Marketing
Itty Bitty Micro Company

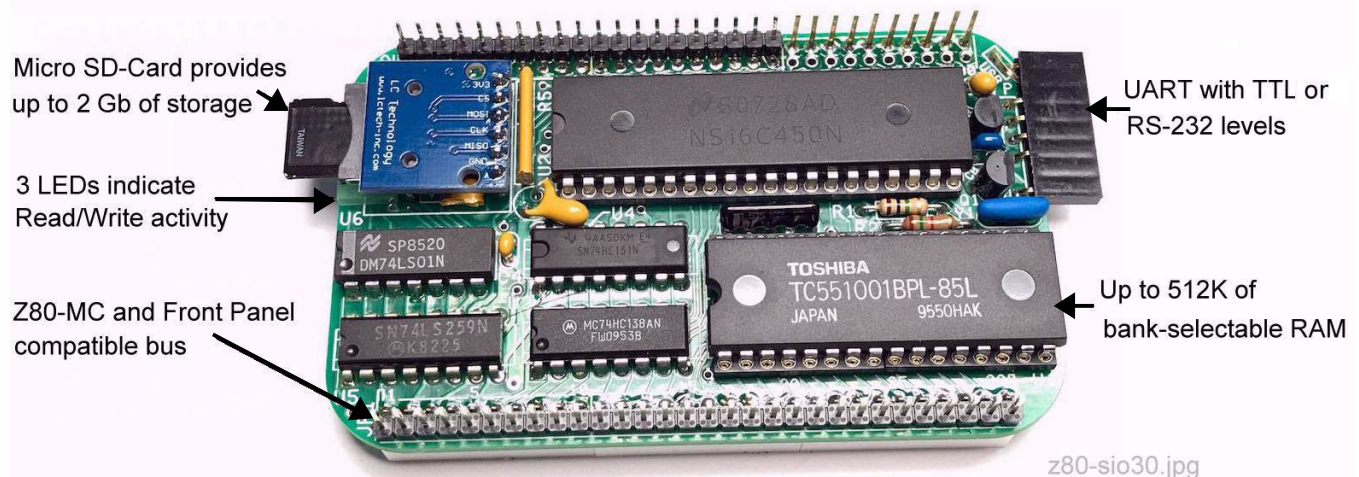
cc: Chip Hacker, hardware herder
Crash Kernigan, codemeister
Wiz Wireman, solderking
Trace Weaver, PCB artiste

Z80-SIO Card - Assembly Manual



Tired of hunting through stacks of floppy disks just to find the program you want? Can't afford a big expensive hard disk? Don't trust unreliable old mechanical contraptions to store your precious data? We have the answer! It's a revolutionary breakthrough in microcomputer data storage. The Z80-SIO Card is the world's smallest disk drive storage system for your Z80 computer. Now you can store **ALL** your CP/M programs and data on one **Itty Bitty Card™**!

We spent over a year designing the Z80-SIO to be the fastest, most powerful, reliable, and affordable solution for mass storage on the market. Put thousands of disks of programs and data (up to 2 Gigabytes) on one completely solid-state card! It's another small miracle brought to you by the micro-geniuses at the **Itty Bitty Micro Company**. Transport your Z80 Membership Card into the 21st century!



Itty Bitty Micro Company, 814 8th Ave N, Sartell MN 56377 USA, (320) 656-9574

leeahart@earthlink.net

<http://www.sunrise-ev.com/z80.htm>

<https://groups.io/g/Z80MC>

Rev.B – last revised 22 May 2024

SUBJECT: Parts List
DATE: 22-Jul-81
FROM: chip8080 [71323,0142]
TO: gilbates241 [74123,0241]

Gil,

Got your manual. It looks good! I guess your hype is no worse than any of the other ads in Byte and Kilobaud. But "micro geniuses"? Is that a compliment or an insult?

The rev.B boards are much nicer; no cuts-n-patches! I'll have Wiz build one up "by the book" (he's WAY faster than me). Then I can play with, err, TEST it carefully.

The manual needs to reflect what we'll actually be sending out. Here's an updated Visicalc parts list to put in the manual. "Source" shows replacement part numbers; but we'll send out what we have on hand. That means some of the parts are a little different than the "source" parts (especially the resistor networks). Here are the substitutions I know of right now.

D2,D3 are zener diodes. They are just being used for their forward voltage drop, so the zener voltage doesn't actually matter. Zeners have a little higher forward voltage drop than regular diodes, which is how I'm getting the 3.3v supply. You might get anything from 1N4734 to 1N4743; they all work the same.

JP1,JP2 are each made of two separate parts; the male pin headers (P1, P2) go on top of the board, and the female sockets (J1, J2) on the bottom.

P1,P2 are the headers. You need a 6-pin, 20-pin, and 30-pin header. I sent two 30-pin parts; cut one of them to make the 20-pin and 6-pin headers. Install the 20-pin and 30-pin headers first, because the female sockets will cover the solder pads for the headers.

J1,J2 are the female sockets for the P1,P2 headers. Two 30-pin parts are required. These weren't available, so I sent six 10-pin sockets to make them (a,b,c). Note that J2c (the socket at J2 for pins 21-30) is BLACK while all the rest are white. The black one needs to be skinny, to clear the 5v regulator on the Z80 CPU board. It also serves as a "key" so you won't plug the boards together backwards.

I included some pictures with notes on them to identify the parts. Throw them in the manual, too. It will help folks identify the parts. Send me an update, and we'll get started on building it!

--

Chip Hacker
Chief (and only) engineer
Itty Bitty Micro Company

| "There ain't no rules around here.
| We're trying to accomplish something!"
| Thomas A. Edison

Z80-SIO CARD -- PARTS LIST				
	A	B	C	D
1	QTY	ITEM	DESCRIPTION	SOURCE
2	---	----	-----	-----
3	2	C1,C2	capacitor 4.7uF 10vdc tantalum	J 94035
4	1	C3	capacitor 4.7uF 25v X5R ceramic 0.1"LS	M FG18X5R1E475KRT
5	1	C4	capacitor 0.1uF 50v X7R ceramic 0.1"LS	D BC3324-ND
6	5	C5-9	capacitor 0.047uF X7R ceramic axial	J 2229811
7	1	C10	capacitor 0.01uF X7R ceramic axial	J 546290
8	3	D1,4,5	diode 1N4148 signal	J 36038
9	2	D2,D3	diode 1N4734 to 1N4743 zener	J 36169
10	1	D6	LED yellow 2x5x7mm rectangular	D 67-1050-ND
11	1	D7	LED red 2x5x7mm rectangular	D 67-1047-ND
12	1	D8	LED green 2x5x7mm rectangular	D 67-1046-ND
13	1	J1	micro SD-card socket	P 31570-MP
14	5	J1abc,J2ab	socket 10-pin Molex KK, 22-18-2101	D WM3241-ND
15	1	JP2c	stacking header/socket 10-pin	J 2177627
16	2	P1,P2	header 30-pin Molex KK series	J 68339
17	*	P1	header 6-pin (* use a piece cut from P2)	
18	1	Q1	NPN transistor with 4.7K/10K, FJN3305R	TMSI (that's me)
19	1	Q2	PNP transistor with 22K/22K, FJN4303R	TMSI (me again)
20	1	R1	resistor 1 Meg 5% 1/4w carbon film	J 691585
21	1	R2	resistor 1.5K 5% 1/4w carbon film	J 690902
22	1	R3	resistor 3.3K 5% 1/4w carbon film	J 690988
23	1	R4	resistor 18K x 4 5-pin SIP bussed	D 4605X-101-183LF
24	1	R5	resistor 10K x 5 6-pin SIP bussed	D CSC10KE-ND
25	1	R6	resistor 330 x 3 6-pin SIP isolated	D 770-63-R330P-ND
26	1	U1	RAM, 32K, 128K, or 512K (128K supplied)	M 913-AS6C1008-55PCN
27	1	U1s	32-pin IC socket for U1	M 575-393632
28	1	U2	UART, 8250 or 16C450 (not 16C550)	J 52610
29	1	U2s	40-pin IC socket for U2	M 575-393640
30	1	U3	74HC138 3-to-8 decoder	J 45330
31	1	U4	74HCT151 8-channel multiplexer	D 296-2139-5-ND
32	1	U5	74LS259 addressable latch	J 47327
33	1	U6	74LS/ALS01 quad 2-in NAND open collector	ebay (or me)
34	1	Y1	resonator 1.8 MHz with capacitors	ebay ZTT1M80MG
35	1	PCB	Z80-SIO PC board, rev.B	TMSI (also me)
36	*	(U3)	27C256 EPROM w.ZMCv1.5 (comes with Z80MC)	TMSI (still me)
37	1	SD-card	micro-SD card, 2 Gb or less, FAT16 format	many sources
38	1	USB	USB-to-SD card adapter/programmer	many sources

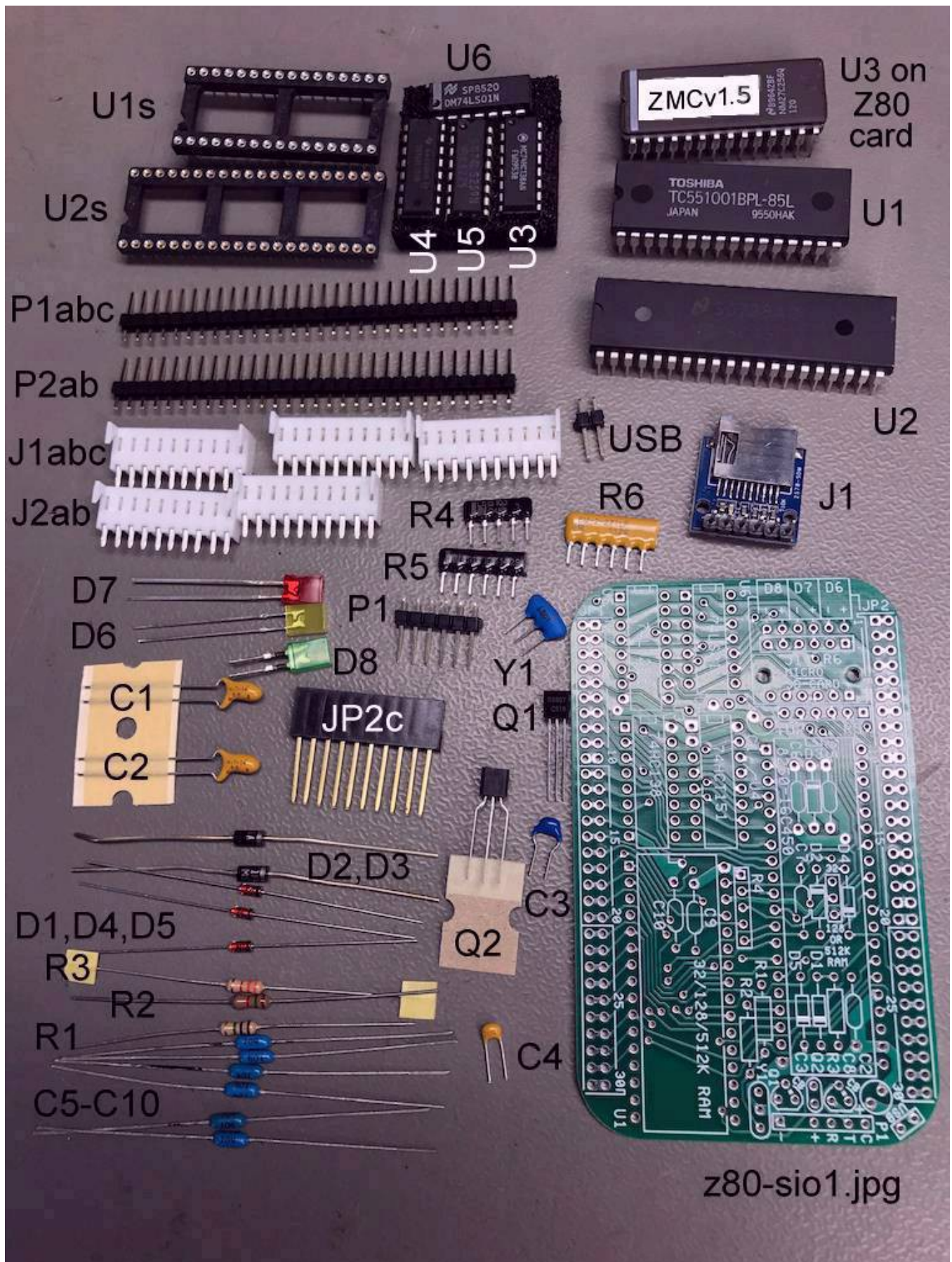
Sources: ebay.com D=digikey.com J=jameco.com M=mouser.com P=mpja.com

Download the files from my web page to your PC. Use the USB-SD card adapter to copy them onto the micro-SD card. This is the easiest way to get your initial CP/M programs and data into your Z80MC.

Also, you need a serial interface to connect to your computer. If it has a traditional RS-232 serial port, you're all set! Build the Z80-SIO card with the RS-232 option. Then all you'll need is wire and the mating connectors to connect the two.

If all you have are USB ports, assemble the card with the TTL option. You'll need a USB-serial adapter. I like the Sparkfun.com USB-serial cable #DEV-9718. It plugs right into header P1 and provides both power and serial I/O. There are others; but there are no standards! Pinouts and pin/socket arrangements are arbitrary. Finding a USB driver that works on your PC may be a challenge as well.

Boards, parts, and complete kits are available from TMSI c/o Lee Hart at www.sunrise-ev.com/z80.htm.



z80-sio1.jpg

SUBJECT: Z80-SIO assembly instructions
DATE: 31-Jul-81
FROM: chip8080 [71323,0142]
TO: wizwireman [72404,1991]

Hey Wiz,

I got the manual back from Gil. I'm sending it to you with a proto kit. See if you can put it together and FOLLOW THE DIRECTIONS this time! I know you can do it blindfolded; but the point is to see if the directions are any good. Take pictures as you go; I'll use them to supplement Gil's incoherent ramblings. :-)

This is the rev.B card: All the changes from rev.A are fixed so you shouldn't have to cut or patch anything. It's pretty crowded -- Tracey had to hide a few parts UNDER the ICs. This means you need to install the parts in the right order, because some parts will block others. Here are some things to watch out for:

Installing the parts

C5-C10, D1-D5 (and R3 if you're adding the RS-232 option) go UNDER the IC sockets. Mount them first, before the sockets.

LEDs D6-D8 mount UNDER the SD-card socket. Install them before the socket. The short pin of the LEDs is the CATHODE (-). Put it in the hole with the "-" sign.

Likewise, SIP resistor R6 gets mounted under the SD-card socket. Bend its leads so it lies flat, with its body toward the middle of the board.

Thinking "Inside the Box"

If you want the 2-card stack to fit inside the Altoids tin (CPU + SIO cards), the boards must be very close together. This takes some extra effort:

1. Don't use normal IC sockets; they are too tall. I sent ultra-low-profile sockets for the UART and RAM chips. If you want to socket the rest, get more of the same kind (digikey.com ED53##-ND, where ## is the number of pins). Or, you can use individual socket pins (digikey.com ED5037-ND) for an even lower height.
2. JP2c: The black 10-pin stacking socket goes in pins 21-30 of JP2. The long pins go on the TOP, and the socket body goes on the BOTTOM of the card. This is necessary to clear 5v regulator U8 on the CPU board. It also acts as a key, so you can't plug the boards together backwards.
3. Here's the hard part: You need to REMOVE the plastic body from the male 30-pin headers at JP1 and JP2 on the top side of the board. I'll describe how to do this later.
4. Now you can install the white 10-pin female sockets. They go on the BOTTOM of the board, with their pins in the INNER row of holes. Plug them onto the Z80MC card, and THEN soldering them to the Z80-SIO card to insure that they line up.

If you DON'T CARE about cramming it into an Altoids tin, build it the "easy way". Just solder the headers on top of the CPU board with their plastic body intact. The cards end up about 0.1" farther apart. This makes room for ordinary IC sockets, too.

--

Chip Hacker
Chief (and only) engineer
Itty Bitty Micro Company

| "There ain't no rules around here.
| We're trying to accomplish something!"
| Thomas A. Edison

0		0
0	SUBJECT: Z80-SIO assembly instructions	0
0	DATE: 8-Aug-81	0
0	FROM: wizwireman [72404,1991]	0
0	TO: chip8080 [71323,0142]	0
0	Hokay, boss. I got the parts and manual. I stuck it all together,	0
0	more or less "by the book". Pictures are enclosed.	0
0	Those headers and white sockets are a real pain. See if you can find	0
0	some more like the black stacking ones (but better quality).	0
0	Now the fun begins. Let me know if it works!	0
0	--	0
0	Willard Wireman	0
0	Senior Technician	0
0	Eastern Coat Rack Company	0
0	"A future to hang your hat on"	0
0		0

Compuserve Email v1.2 <M>enu <N>ext <P>revious <Q>uit <H>elp

SUBJECT: Z80-SIO assembly instructions
DATE: 19-Aug-81
FROM: chip8080 [71323,0142]
TO: wizwireman [72404,1991]

Wiz,

I got your assembled board. It looks great!

I hacked together a manual that pulls together all Gil's stuff, my notes, and your pictures; and sent it to Gil. He put the following assembly instructions together.

We tried to make it as Heathkit-like as possible, with lots of pictures and arrows, and boxes to check as you install each part. It's not finished yet; but at least you can tell if it's on the right track.

Yeah, I know the 74LS01 is hard to find. But I have a whole tube of them to use up.

Note: C5-C10 were blue on the prototypes. The ones we got for production are yellow! So ignore the color in the pictures (electrons are color-blind, anyway).

Look it over, and call me if you see any glaring errors.

Chip

--

Chip Hacker	"There ain't no rules around here.
Chief (and only) engineer	We're trying to accomplish something!"
Itty Bitty Micro Company	Thomas A. Edison

Z80-SIO Card - Assembly Instructions

Here are some Polaroids of the Z80-SIO card during assembly. Insert each part on the top (silkscreened) side of the board. (The only exceptions are the female connectors; I'll point them out when we get to them.) Solder each wire on the bottom of the board. Cut off the excess wire length on the bottom as short as possible, so they won't short to things on the CPU board. As you finish each part, **CIRCLE** the identifier (D4, C5, etc.) so you know each step is done.

D4 1N4148 diode

banded end
on RIGHT

C5 0.047uF yellow, marked 473

D3 1N4734 diode

banded end
on LEFT

C6 0.047uF yellow, marked 473

C7 0.047uF yellow, marked 473

R1 1 Meg resistor, brn-blk-grn-gold

C10 0.01uF yellow, marked 103

D2 1N4734 diode

banded end
on LEFT

C8 0.047uF yellow marked 473

D1 1N4148 diode

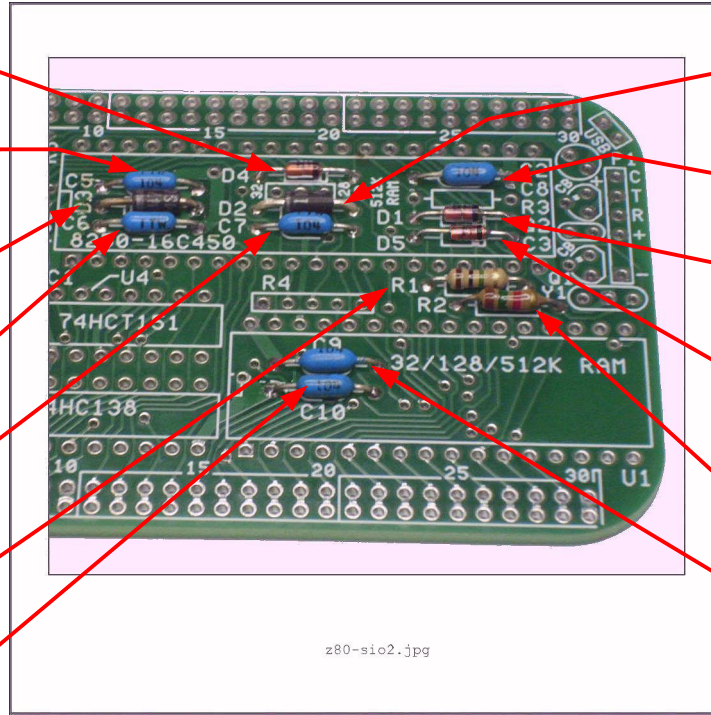
banded end
on RIGHT

D5 1N4148 diode

banded end
on RIGHT

R2 1.5K resistor brn-grn-red-gold

C9 0.047uF yellow marked 473



z80-sio2.jpg

Serial I/O: There are TWO ways to assemble it. Do **ONE** of the following (A or B):

A. TTL serial I/O:

idle: +3v to +5v
active: 0v

Use this option
for a USB-serial
adapter with TTL
or 3.3v levels.
(www.Sparkfun.com
#9718 or equiv.)

USB Short USB to
power the Z80MC
from P1 with a
USB adapter.
Leave USB open to
get power via CPU
or Front Panel.

DO NOT INSTALL
C3, R3, Q1 or Q2.

Q2 short CB.

Q1 short CB.

B. RS-232 serial I/O:

idle: -5v to -12v
active: +5v to +12v

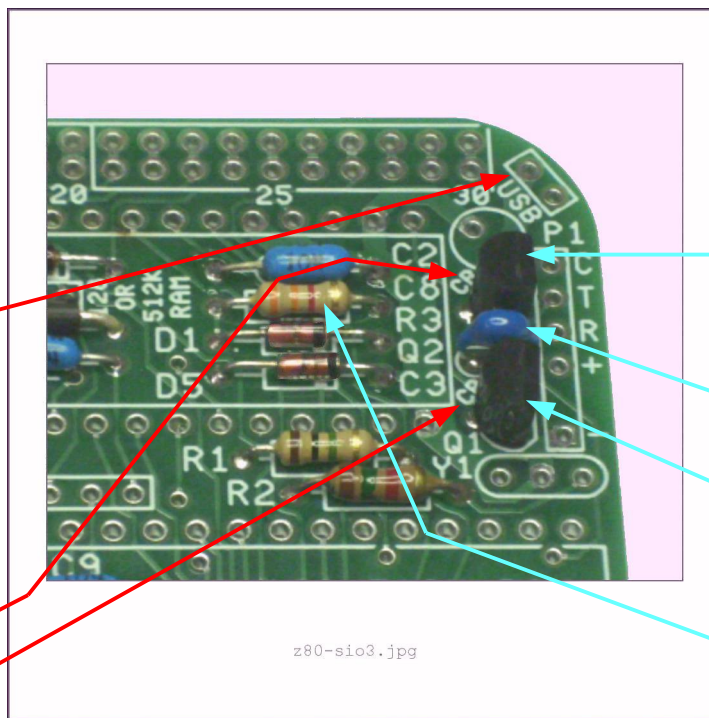
Use this option
for a terminal or
PC that has a real
RS-232 serial port.

Q2 FJN4303
marked R4303,
flat side right,
1/4" high max.

C3 4.7uF capacitor
Blue, marked 475

Q1 FJN3305
marked R3305,
flat side right,
1/4" high max.

R3 3.3K resistor
org-org-red-gold



z80-sio3.jpg

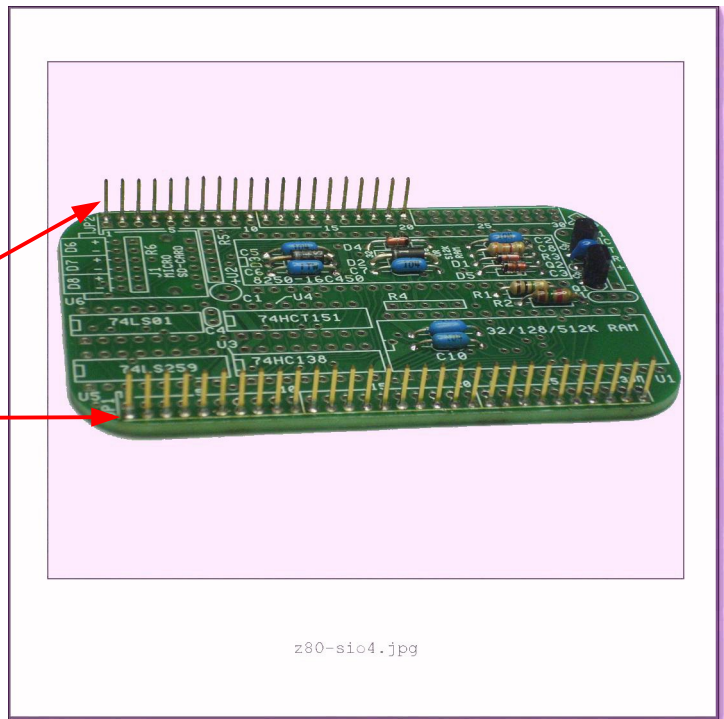
JP1 and JP2: These consist of two parts; the PINS (P1, P2) and the SOCKETS (J1, J2). Install the pins first, because the sockets will block the solder pads for the pins.

The pins go in the OUTER rows of holes, on the TOP side of the board.

P2 20-pin header. Install it along the top edge. Note: The right 10 holes will be EMPTY at this point.

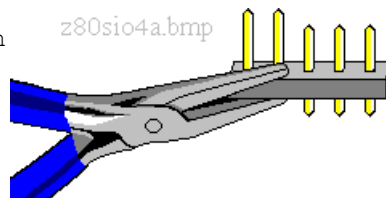
P1 30-pin header. Install it along the bottom edge.

There are TWO ways to do it. If you want the two-card stack to fit inside the Altoids tin, assemble it the "low height" way. Follow instructions #1 to #6. The end result will look like the picture on the right.

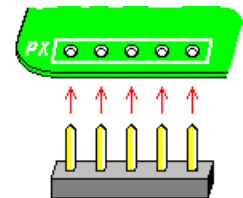


The LOW HEIGHT (hard) way:

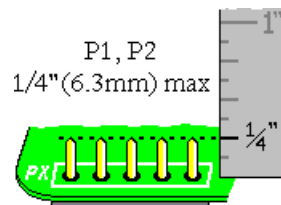
1. Push or tap each pin into the plastic body so it is even with the bottom.



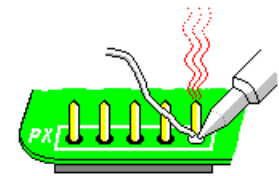
2. Insert the header from the BOTTOM of the board, so the pins poke out on top.



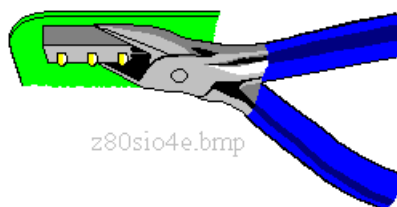
3. Check the pin height above the board. They should be 1/4" (6.3mm) maximum.



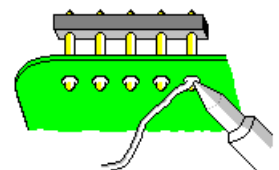
4. Solder each pin on the TOP side of the board. Do not use too much solder!



5. Pry or cut off the plastic body and excess pin length on the bottom of the board.



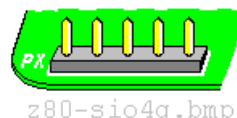
6. If any solder joints look "ugly", push a piece of the plastic body on top to hold the pins in position. Then you can resolder them from the bottom (so any excess solder flows down onto the iron). Remove the plastic body when done.



The FULL HEIGHT (easy) way:

If you don't care if the cards fit inside the Altoids tin, you can do it the "normal" way. This is easier, but the cards wind up about 0.1" farther apart. Follow steps A and B:

- A. Put the header on top of the board, with the short pins in the OUTER row of holes.



- B. Solder each pin on the BOTTOM side of the board.

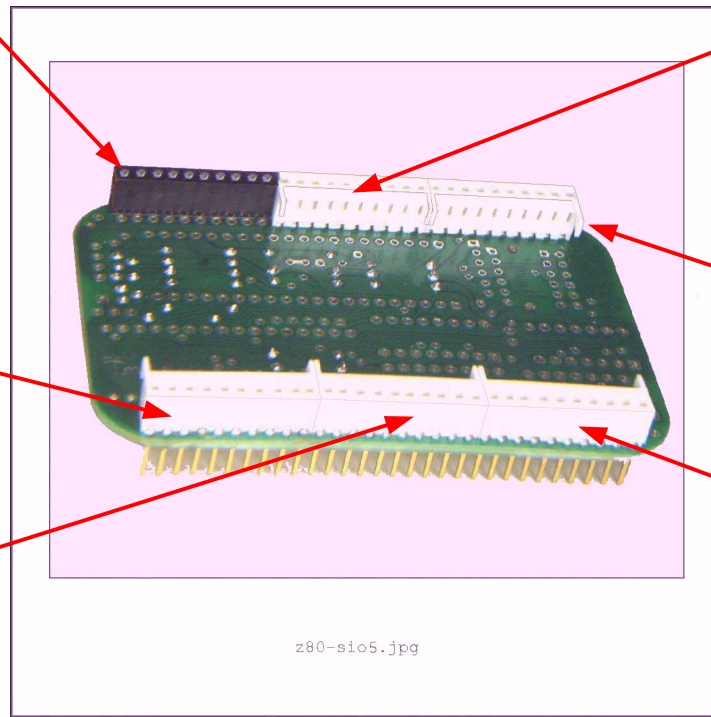
JP1 and JP2: Now install the SOCKETS. They all go on the **BOTTOM** of the board. HINT: Plug the sockets onto your Z80 Membership Card to properly align them. Then put the pins of each socket into the holes on the Z80-SIO card. Finally, solder each pin.

J2c Black 10-pin stacking socket header. Its pins go in the **OUTER** row of holes, at JP2 pins 21-30.

Note: J2c has longer pins. Trim them to match the other pins.

J1c White 10-pin socket. Pins go in **INNER** row of holes "ears" to center, at JP1 pins 21-30.

J1b White 10-pin socket. Pins go in **INNER** row of holes "ears" to center, at JP1 pins 11-20.



J2b White 10-pin socket. Pins go in the **INNER** row of holes, with its "ears" toward the center, at JP2 pins 11-20.

J2a White 10-pin socket. Pins go in **INNER** row of holes, "ears" toward center, at JP2 pins 1-10.

J1a White 10-pin socket. Pins go in **INNER** row of holes, "ears" toward center, at JP1 pins 1-10.

EARS: When finished, cut off the "ears" of the white plastic sockets J1abc and J2ab.

R5 10K 6-pin SIP resistor. Black, marked L61S103. Printed side on left, pin 1 stripe on top.

C1 4.7uF capacitor. Yellow, marked 4u7 10. + lead goes in top right hole marked "+".

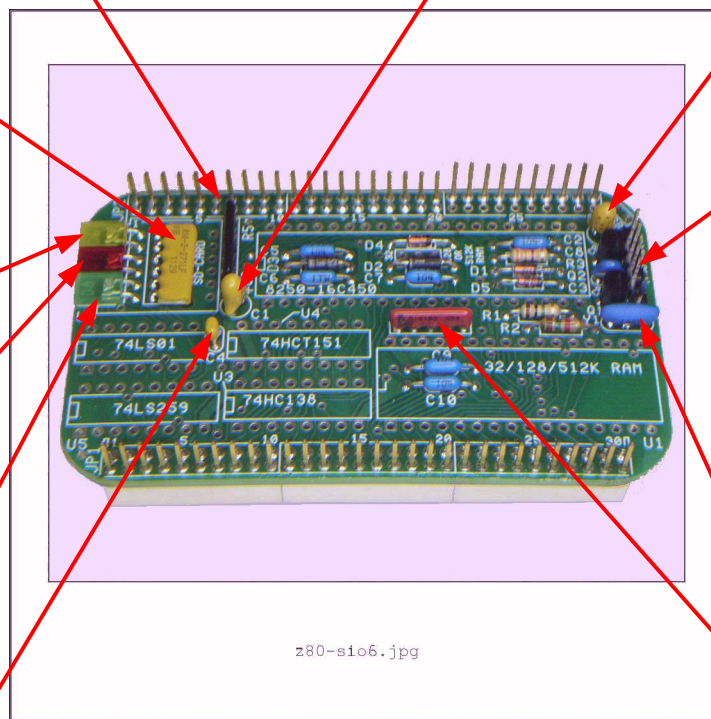
R6 330Ω 6-pin SIP. Black, marked 6B331G. Lay it flat, with the printed side up.

D6 Yellow LED. Long lead goes in hole marked "+".

D7 Red LED. Long lead in hole marked "+".

D8 Green LED. Long lead in hole marked "+".

C4 0.1uF capacitor. Yellow, marked 104.



C2 4.7uF. Yellow, marked 4u7 10. + lead goes in hole marked "+".

P1 6-pin header. Remove 2nd pin from the bottom.

Note: To use P1 with the FRONT PANEL card, bend the pins of P1 to the right, or use a right-angle connector for P1 (photo, page 2).

Y1 Resonator. 3 pins, blue, marked 1.80Y.

R4 18K 6-pin SIP. Marked L51S183. Red, stripe on left, label down.

Integrated Circuits (ICs) are rugged and reliable, but you can blow their brains out with static electricity or careless handling. Handle and install them as follows:

1. Keep the ICs in their protective packaging until ready to use.
2. Straighten any bent pins. ICs usually come with their pins bent out slightly. Set the IC on its side on a table or hard surface, and bend the body so the pins on that side are all parallel to each other, and at right angles to the body. Repeat for the pins on the other side.
3. Prevent damage from static electricity. Hold the IC in one hand, and pick up the board in your other hand. Insert the pins into the holes in the board with the pin 1 end on the LEFT in the photo below. Bend a pin or two to hold it in place until soldered.
4. For ICs that go in sockets, make sure every pin starts into its hole. Then press HARD (the sockets are a tight fit)!

J1 SD-card socket.

Place it on top of the LEDs and R6. Solder, and cut off the extra pin length.

U4 74HCT151.

Pin 1 on left.

U6 74LS01.

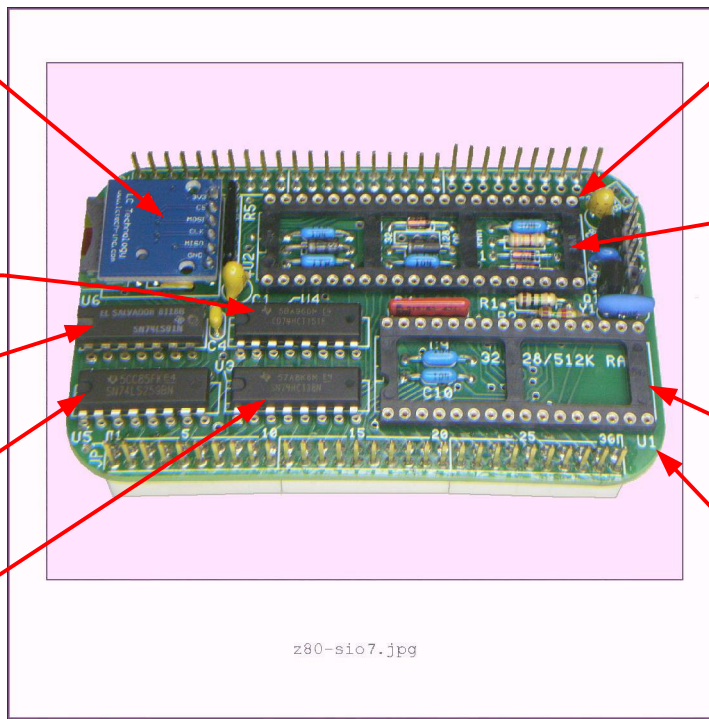
Pin 1 on left.

U5 74LS259.

Pin 1 on left.

U3 74HC138.

Pin 1 on left.



U2s 40-pin socket.

First, install the socket, with the notched end left.

U2: 8250 or 16C450.

Next, install the IC with pin 1 on the left (not shown in photo).

U1s 32-pin socket.

Notched end left.

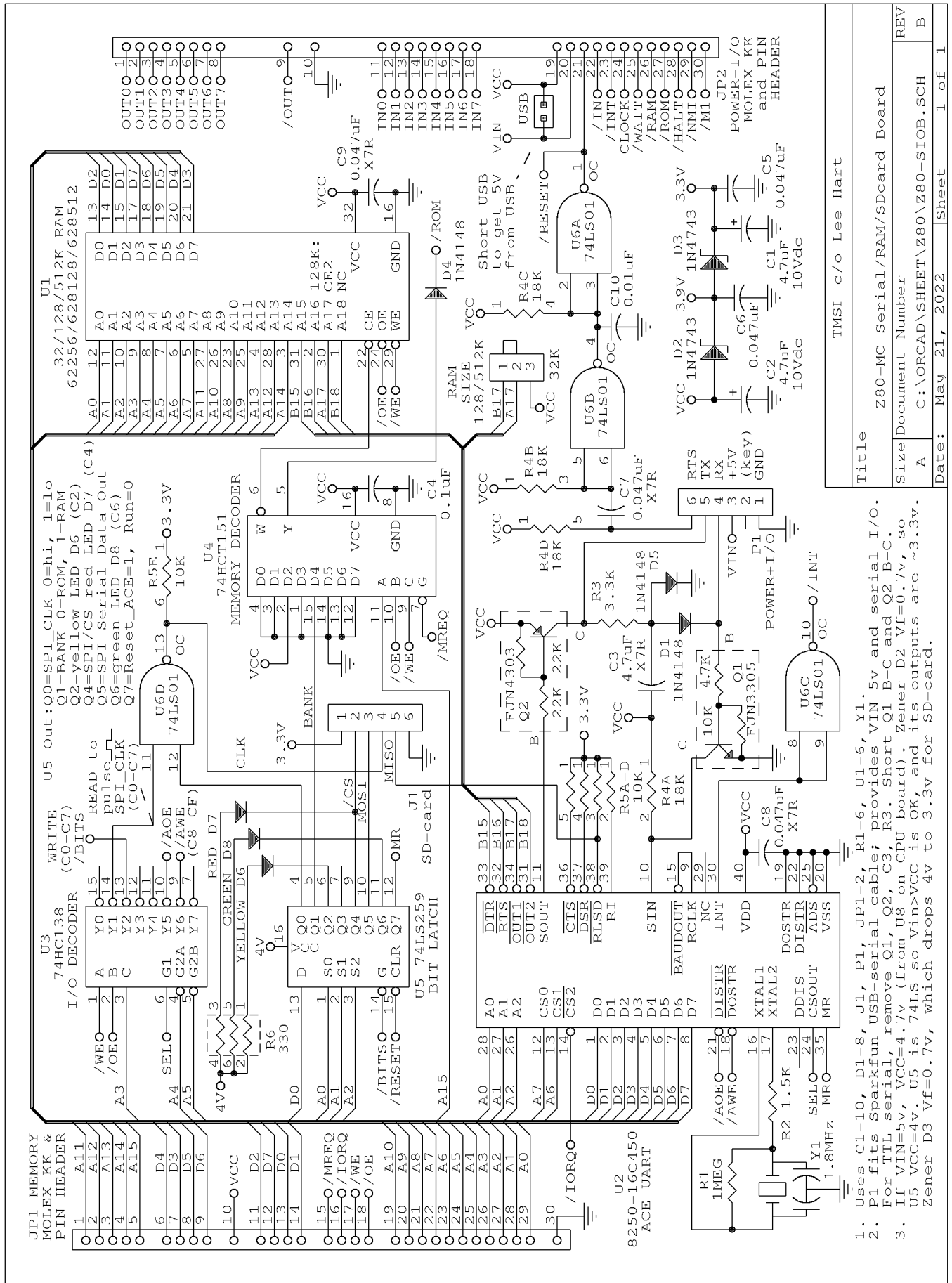
U1 32K, 128K or 512K RAM. 128K AS6C1008 or equiv. supplied. Pin 1 on the left (not shown in photo).

32K/128-512K jumper (under the center of U2): A jumper is already etched into the board for a 128K or 512K RAM. If you want to install a 32K RAM at U1 instead:

- **CUT** the foil jumper (on the bottom of the board) between the center pin and the 128K/512K pin.
- **SHORT** the center pin to the 32K pin.
- The software will need to be changed to match the RAM size installed at U2.

That's it! Your Z80-SIO card is complete. Now download the "Z80-MC Operating Manual with Z80-SIO Card" at <www.sunrise-ev.com/photos/z80/Z80MC-2017.pdf>. You will also find details and updates on the web page at <www.sunrise-ev.com/z80.htm>, such as the files to load onto the SD-card, a Windows CP/M disk explorer program, and more.

I hope you enjoy this little trip into the history of microcomputing. It's a hobby project, and still under development. There are gigabytes of Z80 and CP/M software "out there". But like any good archeological site, it will take some "digging" to discover it, and restore it to usefulness. Have fun, and let us know what you find!



Digital Research CP/M 2.2 Quick Reference – page 1

CP/M is in control when you see the A> prompt. "A" is the current disk drive. Type a Command, any optional filename(s) to use, and the ENTER or RETURN key. Use upper or lower case (CP/M will convert lowercase to uppercase).

CP/M first checks for a built-in command (DIR, ERA, REN, SAVE, TYPE, or USER), then for a .COM file on the current disk (PIP, STAT etc.) If found, it runs it (you don't need to type the .COM part). If not found, it repeats the command with "?" after it as an error message. When the command finishes, it returns to the CP/M A> prompt, ready for the next command.

Each file has a unique FILENAME, with three parts:

d:	disk drive (A-P), followed by a colon (:)
name	then 1-8 alphanumeric characters, including ! @ # \$ % + - (or) but not < > . , : ; = ? * [or]
.extent	a period (.) and 0-3 more alphanumeric characters for the file type. Examples:
.ASM	assembly language source file
.BAS	BASIC program file
.COM	executable program file
.DAT	data file
.DOC	formatted document file
.HEX	Intel HEX format file
.SUB	submit (batch) file
.TXT	simple text file

Filenames can be abbreviated, or use wild cards. "?" matches any single character. "*" matches any string of characters.

D:	for any files on the D: disk drive
FILE	for FILE on the current disk drive
D:FILE	for FILE on the D: drive
FILE.TXT	for FILE.TXT on the current drive
D:FILE.TXT	for FILE.TXT on the D: drive
FILE?	any 5-character name starting with FILE
FILE.*	any file named FILE. with any extent
F*.TXT	any .TXT file starting with F
.	matches everything

CONVENTIONS

{...} Text in curly brackets are optional filenames. (Do not type the curly brackets).

[...] Text in square brackets are "switches" to enable or disable various options. (You must type the square brackets). For example, PIP uses [v] for verify.

CONTROL CODES perform various "control" actions. "^" means hold down the CTRL key, then type the letter. Some control codes may have their own key.

^C	Warm boot; cancel command and re-start CP/M
^G	Ring the console's bell
^H	(Backspace). Backup and delete last character
^I	(Tab). Move right to next tab stop
^J	(Line Feed). Move down a line
^M	(Return or Enter). Process current command, and wait for the next command on a new line
^P	Toggle printer on/off so console output also prints
^Q	Resume a long listing that was paused with ^S
^S	Stop (pause) a long listing
^X	Cancel; backup and delete all characters on line
^Z	Marks end of a file

BUILT-IN commands are part of CP/M (called the CCP), and don't take up any file space on disks. Filenames are optional, and can include wildcards.

B:	change current disk to B
DIR	directory of all files on current disk
DIR B:	all files on B: disk
DIR B:*.COM	all .COM files on disk
ERA file(s)	erase file (or files) on a disk
ERA *.*	erase <u>all</u> files on a disk (caution!)
REN new=old	rename old file to the new name
SAVE n file	save "n" 256-byte blocks of memory starting at 0100h in file
TYPE file	type (display on console) file
USER n	set user number (n=0 to 15)

TRANSIENT Commands are .COM program files on a disk. They can be anything (games, programming languages, word processors, spreadsheets, etc.) CP/M comes with a simple but powerful set, able to write programs, or even rebuild CP/M itself to run on other computers.

ASM - Assembles an 8080 source .ASM file into .HEX and/or .LST files. Use DDT or LOAD to convert the .HEX file into an executable .COM file.

ASM FILE	where FILE is named FILE.ASM and all files are on the same disk as ASM.
----------	---

ASM FILE.SHL	same, but use disks other than A:
S- -	is source disk for FILE.ASM
- H -	is destination disk for FILE.HEX (or Z for none)
- - L	is destination for listing FILE.LST (or Z for none)

Digital Research CP/M 2.2 Quick Reference – page 2

DDT - Dynamic Debugging Tool. An 8080 monitor program to examine and change memory, disk, and CPU registers; and run, trace, or single-step programs. All values are in HEX. Type ^C to exit.

DDT	(no filename) wait for commands
DDT filename	load file, and wait for commands
Astart	Assemble at 'start' address
Dstart{,end}	Dump memory in hex and ASCII
Fstart,end,bb	Fill memory start to end with bb
G{start}{,end}	Go run program at PC, or at optional start address. Optional breakpoint at end.
Hn1,n2	Hex math; show n1+n2, n1-n2
Ifilename	Insert filename in FCB (for R command)
L{start}{,end}	List (disassemble) memory into assembler from start to end
Mstart,end,to	Move start-end block to 'to' addr
R{offset}	Read file (in FCB). No offset overwrites CP/M at 0! Use 0100 etc.
Sstart	Set (change) memory from "start", 1 byte at a time. Type any non-hex character to end.
T{n}	Trace n instructions (show all steps)
U{n}	Untrace n instructions (only show last step)
X	eXamine (show) all CPU registers.
X{register}	eXamine/change selected register
1-bit regs	Carry, Zero, Minus, Even, Interdigit
8-bit reg	A=accumulator
16-bit regs	B=bd, D=de, H=hl, P=pc, S=stack

DUMP - Displays the contents of a file in HEX format. Each line shows the address and next 16 bytes of the file, and continues until the end of the file. DUMP is also supplied in .ASM form as an example program to learn how to write, assemble, and load CP/M programs.

DUMP D:FILE.TXT displays FILE.TXT on D:
 ^S to pause, ^C to exit

ED - Text editor. More like a typewriter than a screen editor, ED commands move an invisible character pointer "p" to perform actions at its location. "n" is the number of characters or lines (the default is n=1), and can be + - or "#" (for all). Separate multiple commands on a line with ^Z.

ED is crude; but powerful. It edits files larger than memory by loading, editing, and saving it a block at a time. It can edit anything, and accepts .SUB commands (with XSUB) for automatic editing.

ED D:FILE.TXT opens FILE.TXT on D: for editing.
If FILE.TXT is not found, create it.

ED commands:

load and save files (read disk -> buffer -> write disk)

nA	Append (read) n lines into buffer (#A for all)
nW	Write n lines of buffer to disk (#W for all)
H	save edited file, clear buffer for more edits
nX	Write n lines to X.SUB file (copy)
R	Read X.SUB file (paste)
E	Exit; save edited file, rename old file .BAK
O	abandon edits, and restart with Original file
Q	Quit without saving

moving the "p" pointer

B	move to Beginning, or -B move to end
n:	move to line "n"
nL	move "n" lines
nC	move "n" characters
nF text	Find nth occurrence of "text" in buffer
nN text	Find nth occurrence of "text" in entire file

displaying and editing text

nT	Type "n" lines (T or 1T is current "p" line)
I text^Z	Insert "text" at "p", and move "p" to end
nD	Delete "n" characters at "p" (-before +after)
nK	Kill (delete) n lines at "p" (-before +after)
nS old^Z new^Z	Substitute (replace) old with new "n" times (-before "p", +after "p", # all)

other commands

nM command(s) ^Z	Macro; repeat command(s) "n" times. 0 or 1 repeats until end of file.
+U	convert to Uppercase, -U don't convert
V	line numbers (+V display, -V don't display)
0V	shows free/total space in buffer

LOAD - Reads an Intel-format .HEX file and writes an executable .COM file. The .HEX file is usually produced by ASM, and set up to run at 0100h under CP/M. The new .COM file can be run by typing its name at the CP/M prompt, like PIP or any other program file.

LOAD B:FILE load FILE.HEX from B: disk, and write FILE.COM to the same disk

MOVCPM - Move CP/M to use a different memory size from 20-64K. (Not used with Z80MC or Altaid 8800.)

MOVCPM n	create "n" kilobyte CP/M system
MOVCPM *	(* or blank) create max size CP/M
MOVCPM **	create max size CP/M, and leave it in memory (ready to save with SYSGEN)

Digital Research CP/M 2.2 Quick Reference – page 3

PIP - Peripheral Interchange Program. Transfers files between disks and peripherals. The format is PIP destination=source1 {,source2...} [switches]

Note: Destination is FIRST; the reverse of the DOS COPY command! Think of PIP as "LET A:=source".

PIP	(no filenames) wait for commands
PIP B:=FILE	copy FILE from current disk to B:
PIP B:=A*.TXT	copy all .TXT files from A: to B:
PIP CON:=FILE	copy (display) FILE on console
PIP NEW=OLD	copy OLD & name the copy NEW
PIP B:=*.COM[V]	copy and Verify all .COM files from current drive to B:
PIP LETTER=HEAD,BODY[T8P]	Copy HEAD & BODY to LETTER, expand Tabs to 8 spaces, and start a new page every 60 lines

The PIP [...] switches are

Dn	Delete characters after column n
E	Echo transfers to console
F	Form Feed (page ejects) removed
Gn	Get file from from User area n
H	Hex data transfer
I	Ignore :00 (null record in HEX file)
L	convert to Lower case
N{2}	Number lines. N2 for "line#:<Tab>"
O	Object (binary) file; ignore ^Z
Pn	new Page every n lines (default60)
Qabc^Z	Quit copying at string "abc"
R	Read hidden .SYS (system) files
Sabc^Z	Start copying at string "abc"
Tn	expand Tabs to next nth character
U	convert to Upper case
V	Verify that copy is written correctly
W	Write over read-only destination file
Z	Zero the high bit (bit 7 or parity)

STAT - Status of disk drives, files, and I/O devices.

STAT filename(s) {Options}
 where options are: \$R/O makes file read-only
 \$R/W makes file read-write
 \$DIR shows file in
 DIRectory
 \$SYS hides from DIRectory

STAT (no filenames) lists free space
 and R/W status of all disks

STAT B: free space and status of B: disk

STAT FILE.TXT size of the file FILE.TXT

STAT *.TXT sorted list & size of all .TXT files

STAT B:.* sorted list, size, SYS (hidden),
 and R/O (read-only) status of all
 files on B:

```
STAT *.COM $R/O    set all .COM files to read-only
STAT DSK:    statistics of all disk drives
STAT USR:    list all user numbers with active files
STAT VAL:    summary of disk and I/O names:
               A: to P:  disk drive names
               CON:     console
               RDR:     reader (serial input)
               LST:     list (printer)
               PUN:     punch (serial output)
```

SUBMIT - Executes a list of commands automatically (like a DOS .BAT file). Commands are listed in a .SUB file, which must be on A:. Prepare this file with a text editor. Use \$1 \$2 ... as placeholders for filenames, which will get filled in by the SUBMIT command line. For example, suppose the MAKE.SUB file contains

ASM \$1
LOAD \$1

Type **SUBMIT MAKE DUMP** to run this .SUB file.
SUBMIT will run ASM and LOAD, replacing each \$1
with DUMP from the SUBMIT command line.

SYSGEN - System Generation. Saves a new copy of CP/M in a disk's "boot" tracks. (Not used with Z80MC or Altair 8800; their RAM disks have no "boot" tracks.)

```

SYSGEN      starts the program
Source Drive (RETURN to skip)
              type drive letter to read boot tracks
              from, or RETURN if they are
              already in memory from MOVCPM
Destination on x: type RETURN
              finish and exit

```

XSUB - extend SUBMIT. When XSUB is the first line in a .SUB file, SUBMIT will read program inputs (as well as file names) from a .SUB file and feed them to the programs executed. Example: if DDT.SUB contains:

```

XSUB
DDT $1
D0100,01FF
G0

```

Type SUBMIT DDT DUMP.COM to run this .SUB file. SUBMIT will run DDT, replacing \$1 with DUMP.COM, then tell DDT to Display the first FF bytes of it (i.e. from 0100-01FF), then exit.

Note: Each line in a .SUB file ends with <CR>, so XSUB can only feed command lines that end in <CR>. Thus, XSUB cannot feed single-character commands not followed by <CR> to other programs.