24 Jan 2021
Rev 3.00.E1

SmallZ80

Assembly and User Manual

Copyright 2019, TG Consulting

Table of Contents

Introduction 5

Record of Changes 6

Chapter 1 – Assembly
- Mods needed for Z180 CPU Adapter 7
- CPU Board 7
- Floppy + 2S Adapter 11
- LAN + ALFAT USB/OEM 15
- Z180 CPU Upgrade 18
- New Floppy + 2S Adapter 18

Chapter 2 - Setup
- CPU Setup / Final Configuration..... 20
- EEPROM Setup 23
- TCP232-E Setup 25

Chapter 3 - Theory of Operation
- CPU Board
  - CPU 29
  - Clock 29
  - EEPROM 29
  - RAM 29
  - General Purpose I/O Port 30
  - Terminal I/O 30
  - Reset Generation 30
  - Memory Map Decoding …..... 31
  - I/O Map Decoding 31
  - Real Time Clock 31
  - Interrupts 31
  - POST-Code LEDs 31
- Floppy + 2S Adapter
  - Floppy 25
  - Serial Ports 25
  - Ethernet LAN Adapter 26
- LAN + ALFAT Adapter
- EEPROM Burner Mode (EEPROM v2.00 or higher)

Table of Contents (cont)

Chapter 4 - Operating System
- Z-System Basics 27
- Stack OS Intro 27
- Stack OS Build Options 27
- Generating a Boot Drive 27
  - CF-IDE 27
  - Floppy 27
- Building Data Drives 28
- CP/M Utilities for SmallZ80 28

Chapter 5 - Programming Considerations
- General Purpose Port 29
- Real Time Clock 29
- LED Bits 29
- External BIOS 30

Chapter 6 - In Case of Trouble
- General Problems 31
- Power Supply Problems 31
- Dead System .....
- XPORT Problems .....
- ALFAT OEM Problems .....
- Interrupt Problems .....
- Terminal Problems .....
- Configuration Setup Problems .....

Chapter 7 - CDROM Contents
- SmallZ80 Software Files 35
- Disk Images 35
- Photos 35

Appendix A - EEPROM POST Codes 37

Appendix B - EEPROM Parameter Passing 38

Appendix C - Expansion Connector 39

Appendix D - Memory Map 40

Appendix E - I/O Map 41

Appendix F - Terminal Serial Connections 42

Appendix G - Updating the BootRom 43

Table of Contents (cont)

Appendix H – Layout and Schematic Diagrams
Mechanical Layout
CPU Mainboard 47
Floppy + 2S 48
LAN + ALFAT USB 49
LAN + ALFAT OEM .....

Schematics
CPU Mainboard .....
Floppy + 2S .....
LAN + ALFAT USB .....
LAN + ALFAT OEM .....
Z180 CPU Adapter .....
New Floppy + 2S .....
New LAN + ALFAT .....

Appendix I – Parts Lists
SmallZ80 Mainboard 63
Floppy + 2S Adapter 65
LAN + ALFAT OEM/USB/SD 66
Z180 Expansion Adapter 67
New Style Floppy + 2S Adapter 68
New Style LAN + ALFAT Adapter 69

Appendix J – Remote Terminal 70

Introduction

The SmallZ80 provides a small, high performance CP/M machine with maximum speed and minimum cost. Tests have shown the full system can run the CPU up to 20MHz without trouble. Terminal baud rates up to 460,800 baud are fully supported, although the terminal might require a significantly lower baud rate. We've found that an above average performing Win10 machine can run a USB-to-Serial connection to the SmallZ80 at 460,800 baud without CTS/RTS handshaking.

The main CPU board provides the user with the following hardware:

- 20MHz Z80 CPU running @ 20MHz
- 32K of EEPROM and 544K of Static RAM (32K fixed, 512K in sixteen, 32K banks)
- Real Time Clock with battery backup
- GIDE interface with dual CF module capability
- System control port with three test code LEDs
- Two serial ports
- Dual Speed CPU clock frequency

A single expansion adapter (Flop2S) is available that adds a dual-drive floppy controller and two additional serial ports. If desired, the serial ports can be connected to a Serial-to-Ethernet module that provides LAN-based connections for Terminal and Printer I/O. The LAN serial ports can operate up to 921,600 baud. Alternately, the terminal can remain on the original serial connection and the LAN connections can be used in other ways.

A second expansion adapter provides a single LAN connection (identical in positioning and layout to the Flop2S adapter) intended for Terminal I/O and a UART connection to the ALFAT OEM, USB, or SD daughterboard adapters. The ALFAT adapters can drive various USB devices, including USB Thumb Drives, with the FAT file system. Software is available to read/write the FAT file system. The ALFAT OEM provides two USB ports and a single SD Card slot; the ALFAT USB provides only a single USB port; the ALFAT SD provides a single SD Card slot.

The CPU PCB footprint is 6.6” wide x 5.0” deep. The serial connectors project from the back side of the board, adding about 1/4” to the depth (although most of the extra would normally project outside the case.) The Dual CF adapter on the GIDE projects nearly 2 3/4” beyond the right side edge, so that must also be taken into account in case selection. The prototype systems use a 9.8” wide x 7.9” deep x 3.2” high case that works quite well and provides adequate space for mounting a single 3.5” floppy drive and external connectors, switches, LEDs, etc.

The CPU board case shown in our website photos is available from JameCo Electronics, Item #209358. The matching, but smaller, case used for 3.5” floppy drives is Item #208929. Both cases require significant metalwork, including cutting, drilling, and filing.

This manual is directed at the vE1 CPU and vB1 Floppy PCBs. Earlier PCB revisions have their own User Manuals. Later revisions will be incorporated into this manual with notes where differences arise between versions and revisions. (The Rev E1 CPU is Version E, Revision 1.)

Record of Changes

Date	Revision
11-15-18	Initial Distribution, CPU vE1, Floppy vB1.
12-19-18	Added comments re: 20/24MHz oscillators and socket. Changed the part #s for the 512Kx8 and 32Kx8 RAM chips to reflect actual #s rather than generic. Changed U18 on CPU from 74LS132 to 74HC132 – it better handles the speed switch. Corrected the description of jumpers J2/4/12/14/15 in the CPU Setup chapter.
03-19-19	Added EEPROM v2.00 details. Continued filling out incomplete sections. Added LAN + ALFAT Rev B0 adapter information. LAN + ALFAT Rev A0 supports only OEM or USB daughterboard; the Rev B0 board adds the SD Card adapter. The PCB is changed only to add the J3 10-pin connector and a mounting hole for the SD Card; the schematic is otherwise unchanged. Reorganized layout, schematic, and parts lists. The User Manual revision now follows the EEPROM code and CPU PCB revision, set at Rev 2.00.E1.
07-23-19	Added EEPROM v2.01 details. Modified Appendix B (Parameter Passing) to reflect v2.01 changes. V2.01 also modified the way the boot software reads the OS into RAM, allowing for a higher base address. Previously, the boot loader worked with a 512-byte granularity, such that if the OS ended at or above FE00H (65024) the entire last 512-byte sector failed to load. Now, the loader uses a 128-byte granularity, just as does CP/M, so the entire OS will load right up to the very top of RAM.
08-05-19	Added EEPROM v2.02 details. Modified Appendix B (Parameter Passing) to reflect v2.02 changes. V2.02 added the ability to have a 'Local' and a 'Remote' terminal. Both are equivalent in terms of the ability to operate the system. See the 'Remote Terminal' Appendix for details. A new CF-IDE image is also required to use the v2.02.xx EEPROM code. In the future, minor EEPROM changes will be reflected in the 'xx' portion of the version, such as 2.02.xx, and will not require a new CF image. Similarly, minor BIOS modifications will also advance the BIOS version from 2.02.00 to 2.02.01, and will net require an EEPROM change. However, if changes to the EEPROM code requires a change to the BIOS code, or a BIOS change requires an EEPROM change, then both version #s will be incremented in the same manner, such as from 2.02.xx to 2.03.00.
11-05-20	Modified design to move I/O ports up to start @ 40H, to allow the Z180 CPU to occupy 00H-3FH. The Z80 CPU version was relocated to maintain I/O compatibility between machines.
12-27-20	Adding Z180 adapter option with new style expansion boards. Document revision changed to v3.00.E1 to reflect major modifications due to Z180 CPU.

Chapter 1 - Assembly

Introduction

The construction process for the SmallZ80 Mainboard changes somewhat when the Z180 CPU board is added. Some of the changes have been incorporated into the initial build of the SmallZ80 to make a later upgrade to the Z180 easier, such as the incorporation of the CPU Riser during the initial build, which eliminates the need to remove the 40-pin socket from the mainboard for the Z180 upgrade.

If you are building directly to a Z180 machine, or converting an existing SmallZ80 to the Z180, the Memory Map Decoder at U14 is not installed. If you might like to revert back to the Z80 at a later date, you can install the socket & bypass cap @ U14.

Other changes, such as the modified I/O Map, have already been incorporated into the firmware and applies to both Z80 and Z180 CPUs.

SmallZ80 Mainboard Assembly

Prior to beginning construction, you should use the CPU PCB as a template to drill the 5 mounting holes (four corners and center) in the case. A 1/8” drill bit works well with the 3mm screws used to mount the PCB. For proper rear-panel alignment with the two serial connectors, the PCB should be ~1/32” from the inside rear panel of the case. Once the holes are drilled, loosely mount the five 3mm x 12mm F-F standoffs to the bottom of the case using 3mm x 6mm machine screws. Mount the PCB to the standoffs with five 3mm machine screws. Once the alignment is straight, snug the screws on top of the PCB. Using a drop of blue thread-locker, tighten the screws on the bottom of the case. Remove the PCB from the inside of the case, leaving the standoffs mounted to the bottom of the case – there should be no reason to remove them.

Read the directions thoroughly before you begin. Construction is straightforward; there are no unusual situations or quirks.

Refer to the silkscreen or mechanical diagram for parts location, arrangement, and orientation. Place the PCB flat on your workbench, silkscreen (or text) up, with the IDE edge connector J10 on the right side, lower corner. In all cases, the pad for pin 1 on the PCB is a square; all other pads are round. If you're building a prototype PCB that does not have the silkscreen legends, you'll have to rely on the mechanical layout diagrams in Appendix A.

- Install IC sockets as follows:
32 pin socket at U4
28-pin sockets at U2 & U3
24-pin (skinny) sockets at U9, U11 & U12
20-pin sockets at U10, U14, U15, U16, & U17
18-pin socket at U8
16-pin sockets @ U6 & U7
14-pin socket at U18
8-pin socket at U13

- Once the sockets are installed and completely soldered, install the 44-pin PLCC @ U5. Note that the socket can be installed 4 directions – three of them are wrong! There is a small arrow in the middle of the socket body – it designates pin 1, which is the square pad on the PCB.

- Install the pair of 1x20 female socket headers in the CPU location @ U1.

- Install 18 .1ufd mono bypass caps at each of the above sockets. Note that all of these caps can have either 5mm or 2.5mm lead spacing. Also install two bypass caps at C13A & C13B near the lower left corner of the PCB. The bypass caps are placed somewhere near the top of each IC (except for C1, which is on the left side of IC1) – refer to the mechanical diagram. Orientation does not matter.

- Clip all excess leads for the bypass caps.

- Install resistors as follows:
    - Nine 10K Ohm resistors (Brn-Blk-Org) at R1, R2, R4, R7/8/9, R14/15/16
    - Six 330 Ohm (Org-Org-Brn) at R3, R5, R6, R10, R17, R18
    - Three 220 Ohm (Red-Red-Brn) resistors at R11, R12, R13
    - Two 6-pin 10K resistor networks at RN1 & RN2 (pin 1 is marked with a dot or stripe)

- Install two 1N4148 diodes at D3 & D4. Note that the stripe on the diode is towards the right side of the board, where it connects to the square pad.

- Install a 1ufd electrolytic cap at C8A. The positive lead goes in the square pad. Bend the leads @ 90 degrees and lay this capacitor flat against the PCB.

- The Y1 oscillator is 20HMz for the Z80 version. If you're building the Z180 adapter, you'll be using an 22.1184MHz oscillator instead. We may be changing the Z80 version to use the same oscillator.

- Install the 14.7456MHz oscillator at Y2. This oscillator is soldered directly to the PCB.

- Install four, 47mfd electrolytic caps at C40, C41, C42, C43. The positive lead goes in the square pad. These capacitors will stand straight up.

- Install the Red (HALT) LED at D1, the Green (PWR) LED at D2, and the Green HiSpd LED at D5. The long lead goes in the square pad.

- Install seven 1x2-pin (non-keyed) headers at J2, J4, J7, J12, J13, J14 & J15.

- Install a 1x3-pin (non-keyed) header at J8.

- Install a 1x4-pin (keyed) header at J1 (Pwr In.)

- Install the 2x15 expansion socket header at Exp.

- Install the on-board push button reset switch at SW1

- Install the eight 1ufd electrolytic capacitors at C6A-C6D and C7A-C7D. The capacitors must all stand straight up. The long pin goes in the square pad. Clip excess leads.

Chapter 1
SmallZ80 Mainboard Assembly (cont)

- Install the 1uf electrolytic capacitor at C13C. It should stand straight up.

- Install the two 9-pin Serial port connectors at J5 & J6

- Install the 40-pin shrouded IDE connector at J10. Be certain of the pin 1 alignment prior to soldering. As usual, pin 1 on the PCB is a square pad. Pin 1 on the header is marked with a small triangle.

- Install the battery holder at B1.

Chapter 1
CPU Board Assembly (cont)

Assembly is complete with the exception of the two 1x6 pin keyed headers at J3 and J9. Prior to soldering these headers into the mainboard, you should consider how you will be using the LEDs and switch that connect to the headers. In all, there are five status LEDs and one (reset) switch connected to the headers. All of the connections on the two headers are intended for front-panel mounted controls and indications, but this is not required and will add a significant expenditure of time and effort.

The connections on J3 provide front panel display of Power and IDE activity, along with the Reset switch. There is a Reset pushbutton switch mounted on the mainboard, so the front panel switch is not immediately necessary. Similarly, there is a power LED on the mainboard, so the front panel power LED is also redundant. The front panel IDE LED is the only IDE activity indication on the SmallZ80, although the CF-IDE adapter also has an activity display LED (but may be difficult to see.)

J9 connects only the three status LEDs in Red, Green, and Yellow. These are used by the system to display certain status information. Refer to Appendix B – EEPROM Post Codes for details.

Basically, neither of the two headers must be installed for proper computer operation. It might be desirable for you to take a 'quick and easy' approach at first. Later, when you've determined the best way to proceed, you can then take the right approach for your needs and the requirements of the case in which you install the SmallZ80.

You can, for instance, solder the 5 LEDs directly to the mainboard, without the header or wired connectors. The sixth connection would be for the front panel reset switch, but that can be handled temporarily by the mainboard-mounted reset switch.

The J3 header connector has the following connections (and wire colors):

```
1 – Red – Power LED + (long lead on Grn LED)
2 – Blk – Power LED Gnd (short lead, flat side on Grn LED)
3 – Blu – Reset (connects to either side of Reset button)
4 – Blk – Reset Gnd (other side of Reset switch)
5 – Gra – IDE Activity LED + (long lead on Yellow LED)
6 – Blk – IDE Activity Gnd (short lead, flat side on yellow LED)
```

J9 connects to the three POST Code LEDs (wire colors match the LED color):

```
1 – Red – Red LED + (long lead)
2 – Blk – Red LED Gnd (short lead, flat side)
3 – Yel – Yellow LED + (long lead)
4 – Blk – Yellow LED Gnd (short lead, flat side)
5 – Grn – Green LED + (long lead)
6 – Blk - Green LED Gnd (short lead, flat side)
```

This completes the CPU assembly. Do not install any ICs until after the power-on testing is complete in "SmallZ80 CPU Setup / Final Configuration."

Chapter 1
Floppy + 2S Adapter Assembly

The Floppy + 2S Rev B1 Expansion adapter can be assembled in two sections: Floppy and Serial ports. There is no interaction between the two sections; you can build either without the other. Additionally, the Serial ports can be used as normal serial ports, or they can drive the TCP232 LAN module. Both functions may be present on the adapter, but only one can be used at any given time. The Ethernet connection requires the installation of two SMD 74LVC125 SOIC chips; these are already installed and will have no effect on the serial ports without the optional 'LAN Connection Kit'.

Read the directions thoroughly before you begin construction. Construction is straightforward; there are no unusual situations or quirks.

Refer to the silkscreen or mechanical diagram for parts location, arrangement, and orientation. Place the PCB flat on your workbench, silkscreen (or text) up, with the Floppy edge connector J1 on the right side, upper corner. In all cases, the pad for pin 1 on the PCB is a square; all other pads are round. If you're building a prototype PCB that does not have the silkscreen legends, you'll have to rely on the mechanical diagram later in this document.

Before beginning the construction, remove the CPU board from the case and install three 4-40 x 1/2” F-F hex standoff posts in the PCB, using the holes in the Floppy + 2S adapter to determine which holes are used on the CPU board. Leave the 4-40 machine screws slightly loose for now. Do not re-install the CPU board into the case.

Floppy Disk Controller:

- Insert and solder the PLCC-64 FDC socket. There are four ways to install the socket into the PCB – three of them are wrong! Note the arrow in the inner section of the socket. It points to pin 1, which is the square pad on the PCB.

- Install the .1uf mono caps at C1A, C1B and C1C spaced around the FDC socket.

- Install the 6-pin, 220 network resistor (A221) at RN1. Pin 1 is marked with a dot or stripe.

- Install the 6-pin, 10K network resistor (A103) at RN2. Pin 1 is marked with a dot or stripe.

- Install the 24MHz crystal at X1.

- Install two 10pf mono caps at CX1 & CX2.

- Install the 34-pin box header at J1. Pin 1 is marked with a small triangle on one of the long sides of the header .

Chapter 1
Floppy + 2S Assembly (cont)

UART I/O - The UART at U2 is required for either Serial or LAN I/O.

- Install the 44-pin PLCC socket at U2. Note the arrow inside the socket - it points to pin 1, which is a square pad on the PCB.

- Install the 14.7456MHz crystal at X2.

- Install the .1uf bypass cap at C2.

- Install the two 10pf mono caps at CX3 & CX4.

- Install three, 3-pin headers at J3, J4, & J5. There are three shorting blocks that must be installed on these headers. The position they're installed in depends on whether you are building for a LAN connection or dual serial ports. Install all three shorting blocks in the same position, either "L" for LAN or "S" for Serial.

2-Port Serial I/O – For standard serial ports.

- Install two 16-pin DIP sockets at U3 & U4.

- Install two .1uf bypass caps at C3 & C4.

- Install the 2x10 pin header at J2. Note the arrow on the header body that denotes pin 1 and ensure it aligns with the square pad on the PCB.

- Install eight 1uf electrolytic caps at C31-C34 and C41-C44. The long lead (+) goes in the square pad.

TCP232 Ethernet LAN

- The two 74LVC125 SOIC SMD chips should already be installed at U12 & U13.

- Install two .1uf bypass caps at C12 & C13.

- Install the 3.3v regulator at U14. The flat side goes against C12 & C14.

- Install the 47uf electrolytic cap at C14. The long pin goes in the square pad. This component is too tall to stand upright; lay it flat along the edge of the PCB.

- Install the 5-, 7-, and 9-pin socket header strips for the TCP232 daughterboard. Fit the headers to the TCP232 module, then insert the module into the PCB. Holding the module in place, solder the socket headers to the PCB. This method provides perfect alignment of the headers.

- Install the 2-pin header at J6.

Chapter 1
Floppy + 2S Assembly (cont)

Common to All Configurations

- Install two 47uf electrolytic caps at C10 & C11. The long (+) lead goes in the square pad.

- Install (but do not solder) the 2x15 expansion pin header on the bottom of the PCB, so it will mate with the 2x15 pin socket header on the CPU board. Insert the header into the corresponding female socket header on the CPU board. Using three 4-40 machine screws, loosely mount the adapter to the standoffs on the CPU board. When all parts are in good alignment, tighten the screws on the bottom of the CPU board. (These screws will remain in place – we recommend you use a drop of blue thread locker on each screw.) Snug up the two screws on top of the Flop2S adapter. Ensure the pin header is pressed up against the bottom of the Flop2S and solder the pins on the top of the adapter board.

This completes the Floppy + 2S assembly.

Chapter 1
LAN + ALFAT OEM/USB Adapter

The LAN + ALFAT adapter can be assembled in three sections: Common, LAN, and ALFAT. There are no interactions between the LAN and ALFAT functions; you can build either without the other. Since the serial ports are separate, both functions may be present on the adapter and used at the same time. The Ethernet connection requires the installation of one SMD 74LVC125 SOIC chip; the ALFAT uses a single SMD 74VHC04 SOIC chip; both chips are already installed.

Read the directions thoroughly before you begin construction. Construction is straightforward; there are no unusual situations or quirks.

Refer to the silkscreen or mechanical diagram for parts location, arrangement, and orientation. Place the PCB flat on your workbench, silkscreen (or text) up, with the expansion edge connector Exp on the right side, lower corner. In all cases, the pad for pin 1 on the PCB is a square; all other pads are round. If you're building a prototype PCB that does not have the silkscreen legends, you'll have to rely on the mechanical diagram later in this document.

Before beginning the construction, remove the CPU board from the case and install two 3mm x 12mm F-F hex standoff posts in the PCB, using the holes in the LAN + ALFAT adapter to determine which holes are used on the CPU board. The lower left and upper right round pads are the correct holes on the expansion adapter. Leave the machine screws slightly loose for now. Do not re-install the CPU board into the case.

Assembly common to both configurations:

- Verify that the two SOIC SMD chips are installed at U2 & U3. If not, they will be in your kit and should be installed first.

- Install the PLCC-44 UART socket at U1. This socket can be placed in four orientations – three f the are wrong! The inner surface of the socket has an arrow pointing to pin 1 – it must point towards the square pad on the PCB. Additionally, the upper left corner of the socket is beveled, as is the silkscreen layout on the PCB.

- Install .1uf bypass caps at C1 and C4. One lead of the capacitor must be in the leftmost hole; the second lead will fit in either the center or rightmost hole.

- Install three 47uf electrolytic caps @ C10, C11, and C12. The long (+) lead goes in the square pad.

- Install a 10K resistor @ R1.

- Install the 14.7456MHz UART crystal at Y1. Be sure the case is raised off of the PCB by a small amount to avoid shorting on the via between the pins. A small piece of masking tape will suffice.

- Install two 22pf monolithic caps at CX1 & CX2.

Chapter 1
LAN + ALFAT OEM/USB Adapter (cont)

- Install the 3.3v regulator @ U4. You should use a small dab of heat conducting paste between the IC and the PCB. Use the supplied 4-40x3/8” machine screw and nut to mount the regulator, then solder it in place.

- The only remaining 'common' component is the expansion connector at Exp. It is installed from the bottom of the PCB and will interfere with the rest of the assembly. It will be installed when all other component installation is complete.

TCP232 Ethernet LAN

- The 74VLC125 SOIC at U2 was installed earlier. The 74HC04 SOIC at U3 is not used by the LAN adapter.

- Install the .1uf bypass cap at C2. One lead of the capacitor must be in the leftmost hole; the second lead will fit in either the center or rightmost hole.

- Install the -5, 7-, and 9-pin socket header strips for the TCP232 daughterboard. Fit the headers to the TCP232 module, then insert the module into the PCB. Holding the module in place, solder the socket headers to the PCB. This method provides perfect alignment of the headers.

- Remove the TCP-232 LAN daughterboard and set aside. It will be re-installed when power checks are complete.

- Install the 2-pin header at J3.

This completes the assembly of the LAN section. If you are not installing the ALFAT OEB/USB adapter, skip down to 'Final Assembly.'

ALFAT OEM/USB

The ALFAT adapter requires that both the SOIC chips be installed @ U2 & U3. They should have been installed earlier.

The ALFAT section of the PCB allows for either the ALFAT OEM or the ALFAT USB. The installation of the 2x5-pin connector at either J1 or J2 is determined by which ALFAT is being used. Your ALFAT adapter should already have the male 2x5 pin extension installed on the bottom of the ALFAT PCB.

- Install the .1uf bypass cap at C3. One lead of the capacitor must be in the leftmost hole; the second lead will fit in either the center or rightmost hole.

- Install the 2x5 pin connector at J1 (for the ALFAT USB) or J2 (for the ALFAT OEM.)

Chapter 1
LAN + ALFAT OEM/USB Adapter (cont)

- Loosely mount - FROM THE BOTTOM - four (OEM) or two (USB) 12mm standoffs with the supplied 6mm machine screws, one at each corner of the ALFAT OEM (or two at the top of the ALFAT USB.) Inserting the bottom-mounted male pins into the appropriate socket on the Expansion adapter PCB, mount the ALFAT to the Expansion board with four (OEM) or two (USB) machine screws. The 2x5 pins and socket will force the proper positioning of the ALFAT daughterboard on the standoffs. Once the daughterboard is in place, you can snug down the machine screws on the top of the adapter. The screws at the bottom of the adapter will not need to be removed in the future, so you should use a drop of blue thread-locker, then tighten the bottom mounting screws.

- Remove the ALFAT from the Expansion adapter and set aside. It will be re-installed when power checks are complete.

LAN + ALFAT Final Assembly

The only remaining assembly step is to install the 2x15 expansion connector on the bottom of the LAN + ALFAT Expansion adapter. The Mainboard should already have the 2x15 female expansion socket installed. Installing from the bottom of the Expansion adapter, insert the 2x15 male expansion header (do not solder yet.) Insert and snug down the top screws mounting the adapter in place and do the same for the screws on the bottom of the Mainboard. With the Adapter fixed in place, ensure the expansion pins project above the Expansion adapter and are snug against the bottom of the Adapter. Solder the pins.

The bottom side machine screws should not have to be removed in the future, so a drop of blue thread-locker on each screw will ensure they remain snug.

This complete the LAN + ALFAT Adapter construction.

Testing

There are only two power checks to be made on the adapter: +5v and +3.3v. Re-install the Mainboard into the case and install the adapter in place on the Mainboard. Power up the SmallZ80. With a multimeter, read the voltage on Pin 1 of the TO-92 regulator at U4 (the negative lead can connect to one of the mounting screws.) Pin 1 should be at +5vdc. Switch the + lead to pin 3 of the U4 regulator. It should read +3.3vdc.

- Power down the SmallZ80. If the checks were good, continue with the Final Assembly below. If the tests failed, stop and determine the cause. Don't hesitate to contact TerryG@stack180.com for assistance.

Chapter 1
LAN + ALFAT OEM/USB Adapter (cont)

Final Assembly and Testing

- Install the 16C2550 UART at U1. Note that one of the beveled edges of the UART has a small dot – this indicates pin 1, which is in the middle of the top edge of the UART socket.

- The LAN daughterboard fits into the riser headers. Once in place, you can use the included tie wrap to hold it in place. Since the LAN daughterboard projects beyond the edge of the Mainboard (and through the back side of the case), you may be installing/removing the daughterboard several times before you're finished. To facilitate the removal, you might want to hold off on installing the tie wrap. The tie wrap is passed down the right side of the daughterboard, through the hole on the PCB Expansion PCB, then up around the left edge of the PCB and back through the tie wrap to lock the daughterboard in place.

- The ALFAT OEM (or USB) is mounted with four (or two) machine screws. This daughterboard is flush with the back side of the Mainboard, so you should not have to remove the ALFAT from the Expansion adapter.

One final test remains to verify that the CPU is communicating with the Expansion adapter. With the Adapter installed, power up the SmallZ80 and check the sign-on display. You should see something like this:

```
SmallZ80 RevE Monitor ROM v2.00          Terry's SmallZ80 Testbed
  Copyright 2019 TG Consulting
  Compiled 03/01/19        **Private**
  Banked RAM Capacity:  16 banks of 32K present (EEPROM Shadowed in RAM)
  Software Data Prot:   TRUE
  Initializing GIDE:    Master: 993x16x16       Slave: 980x16x16
  Expansion Adapter:    LAN+ALFAT
  ALFAT Startup:        On-Line & Clock Set!

20:27  SmallZ80 (H=Help) :
```

Note the 7th line: "Expansion Adapter: LAN+ALFAT". If you don't get that line, the CPU is not detecting the Adapter properly. Contact TerryG@stack180 for assistance. If your adapter includes the ALFAT daughterboard, you will also get the 8th line as shown. If the ALFAT daughterboard is not present, you'll see "ALFAT Startup: TimeOut!" This is normal without the ALFAT.

This completes the LAN + ALFAT assembly and testing.

Chapter 1
Z180 CPU Upgrade

Chapter 1
New-Style Floppy + 2S Adapter

The new-style Floppy + 2S adapter is very similar to the original, but there are some significant enhancements. The optional components were moved off-board, so the entire board is assembled as a unit rather than as separate, optional sections.

Read the directions thoroughly before you begin construction. Construction is straightforward, but there are a couple of quirks. First, C10 is not used – installing an electrolytic capacitor at C10 will prevent the TCP232 connector from being installed in the 8-pin housing. Second, C11 should be installed on its side. Due to the arrangement of components, the leads need to be bent at 90 degrees such that the body of the capacitor points towards the bottom of the PCB, obscuring part of the "SmallZ80" silkscreen legend.

Refer to the silkscreen or mechanical diagram for parts location, arrangement, and orientation. Place the PCB flat on your workbench, silkscreen (or text) up, with the Floppy edge connector J1 on the right side, upper corner. In all cases, the pad for pin 1 on the PCB is a square; all other pads are round. If you're building a prototype PCB that does not have the silkscreen legends, you'll have to rely on the mechanical diagram later in this document.

The build is nearly identical to the original Flop+2S. Follow the original instructions for the Floppy and Serial sections, modified as follows:

– The 8 electrolytic capacitors are replaced by Tantalum caps.

– J6 (8-pin right angle connector) is a new component. Be sure you do not install C10 or it will conflict with installing the LAN cable into J6.

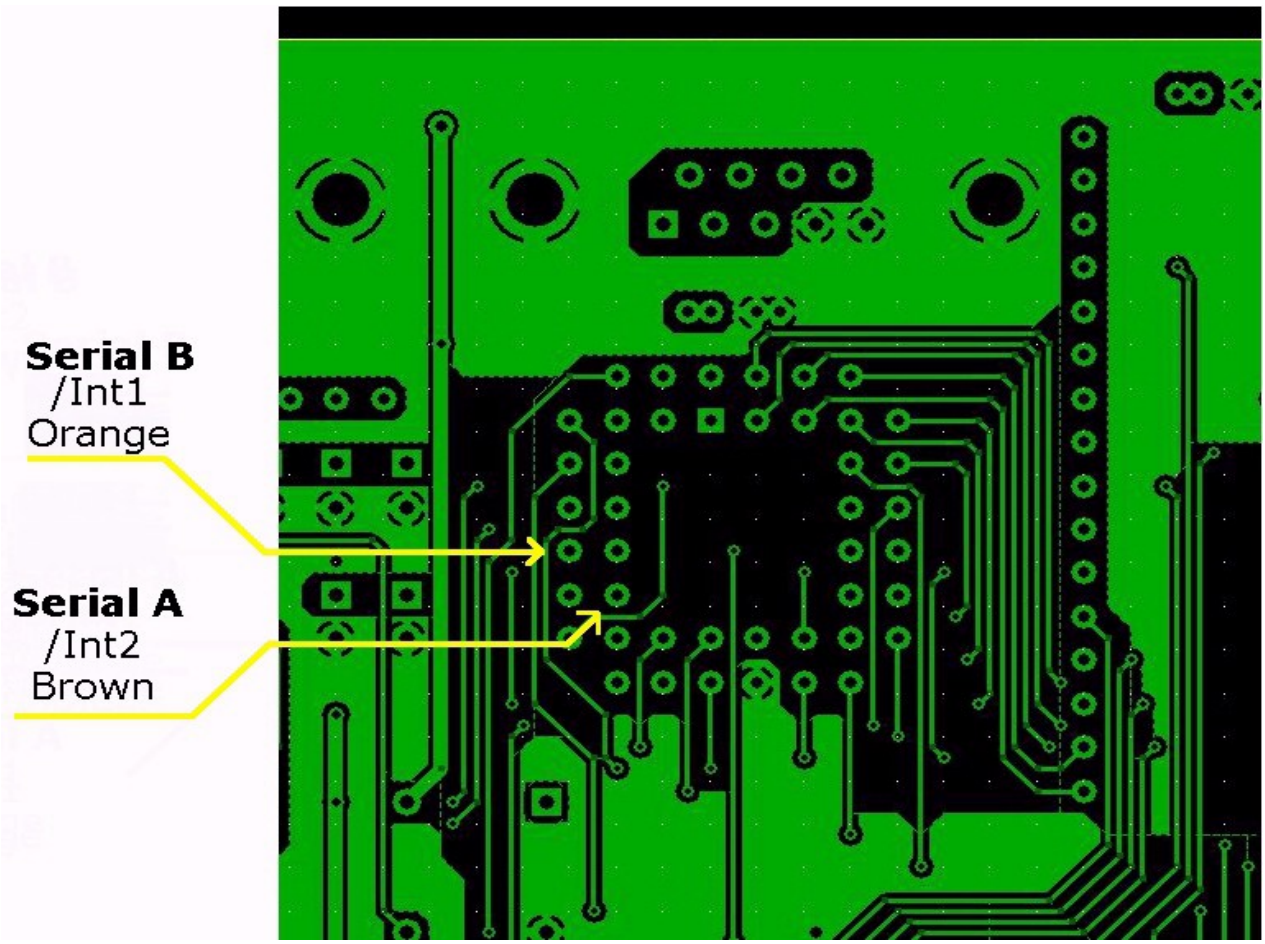
– The Expansion connector is installed from the top of the board.

– J3/4/5 are used to determine which port is sending signals to the UART. When the three shorting blocks are in the upper (1-2) position (L), the LAN adapter is sending signals to the UART; with shorting blocks in the lower (2-3) position (S), the MAX232 serial level-converters are passing data to the UART. J3 and J4 are for Serial-D while J5 is for Serial-C. You can split the signals such that Serial-D is connected to the LAN (J3 & J4 in the top (L) position,) and Serial-C is connected as a standard serial port (J5 in the lower (S) position.) You can revers those connections, but generally, if you're going to the use the LAN adapter with one connection, it is best to do so with Serial-D. (Serial-D has handshaking connections to the TCP232 LAN adapter; Serial-C does not.)

– The remote LAN adapter daughterboard connects to J6 via an 8" cable (supplied with the LAN adapter.) The daughterboard can be mounted anywhere within reach, with the RJ45 LAN connector open through the case's rear panel.

Chapter 1
New-Style Floppy + 2S Adapter (cont)

– J7 (Int Out) and J8 (Int In) are new. J8 receives the UART interrupt signals from the mainboard. They are 'ganged' together with the on-board UART interrupts and J7 passes them to the Z801 CPU's Int In connector. The 12” Org/Brn cable is connected to the mainboard UART as shown here:



This is a back-side view of the mainboard, centered on the 16C2550 UART. Note the connections for the Brown and Orange wires. These wires can be passed through from the top of the mainboard at the (unused) mounting hole immediately to the left of the RTC @ U8.

Chapter 2
CPU Setup / Final Configuration

The case will require some work before you're ready to power up the system:

- Drill a hole and mount the Power input connector. The power input connector must match the output connector on the wall wart power supply. We've used a single 2.5A wall wart to drive the entire system, including two floppy drives.

- Wire power to the On-Off switch

- Connect switched power to the 4-Pin power input connector on the CPU board. The connector is keyed – be sure that you connect the power supply '+' side to the red leads!

NOTE: The next two steps - connecting the two 6-pin connectors - are not necessary for the initial testing. The wires on the connectors should be of sufficient length to accommodate your case layout.

- The J6 and J9 6-pin connectors are discussed in some detail on the last page of CPU construction here.

- Once you've made the above case modifications and BEFORE installing the ICs, we suggest you do a Power On test to verify power supply operation safely. Mount the CPU board in the case and connect the power supply. Verify 5V at several locations around the PCB (refer to the schematics and PCB mechanical layout diagrams in this document.)

- Remove power and install 18 ICs as described in the part list and the mechanical drawing.

- B1 is the RTC Battery - install the CR2032 battery here. Do this before installing the battery jumper @ J7.

- There are three configuration jumpers that must be placed as required:

J7 Used to supply battery power to the RTC. When the shorting block is installed, the RTC will receive battery power when main power is removed; the battery keeps the RTC powered and timekeeping will continue. Install this jumper after the battery is installed in the holder. The battery will be good for several years.

J8 Used to Write Enable or Write Protect the EEPROM. This jumper must be on pins 2-3 (WE) to allow the Setup data table to be written. It is safe to leave it installed for Write Enable and we recommend you do so. Jumper on pins 1-2 will Write Protect the EEPROM.

J13 Supplies +5v to Pin 20 of the IDE port @ J10. This is used to provide power to a CF-IDE adapter the recognizes Power on Pin 20. **You must not have removed Pin 20 from J10 for this jumper to be used.**

Chapter 2
CPU Setup / Final Configuration (cont)

- There are five additional jumpers used to control various aspects of system operation and configuration:

J2 Normally left open. Installing a shorting block on J2 will force the system to connect the terminal on Serial-A @ 9600 baud, HFC disabled, regardless of the Setup or prior configuration settings.

J4 Normally left open. Installing a shorting block on J4 will force the system to stay in low speed on the UART clock @ 14.7456MHz, rather than shifting to the high speed clock (20MHz) during the initialization cycle.

J12 & J14 Undefined.

J15 Normally left open. Installing a shorting block on J15 will force the system into EEPROM Burner mode, where you can burn an EEPROM installed in the 32K Ram socket or re-burn the Main EEPROM. (EEPROM v2.00 or higher required.)

The CP/M OS is delivered on a 64MB or 128MB CF module, with a dual CF-IDE adapter. Insert the CF into the adapter, face up on the side of the adapter with the LEDs. This is the 'top' side. Then, insert the CF-IDE adapter into the IDE port, 'top' side FACE DOWN. The CF-IDE adapter and the IDE socket are keyed - be sure you match up the keys. Carefully check the placement of the adapter - it can easily be offset by one pin. Install a shorting block on J13 to supply power to the CF-IDE adapter (if pin 20 was retained. If you removed Pin 20, you'll have to supply power to the CF-IDE via a 3.5” disk drive power connector.)

- Establish the connection to your terminal via the Serial-A DTE port. Set your terminal to 9600 baud, no Flow Control, 8N1. Most terminals, and all PCs if using terminal emulation software, have DTE ports, just like the port on the SmallZ80. To make the DTE-DTE connection, you're required to use a NULL Modem adapter or cable, which will make the connection act like DTE-DCE instead of DTE-DTE (which will never work.) We've found that a USB-to-Serial connection cable is the best connection possible. They allow baud rates well in excess of 115,200 baud and they can do so without handshaking. We use USB-TO-SERIAL (Male) cables with a null modem female-to-female slimline data transfer adapter to make the connection. There are also USB-TO-SERIAL (Female) null modem cables that will connect directly without the need of a slimline adapter (although the one we tried worked, it did so only when handshaking was enabled.)

Chapter 2
CPU Setup / Final Configuration (cont)

- Power up the SmallZ80. You should get a sign-on message on the terminal. If it's garbled, a baud rate mismatch is likely. If nothing shows up, reset the CPU and ensure the terminal has actually opened the connection. The sign-on message will be very similar to this:

```
SmallZ80 RevE Monitor ROM v1.00     SmallZ80 Example Owner String
  Copyright 2018 TG Consulting
  Compiled 10/29/18     Distribution
  Banked RAM Capacity:  16 banks of 32K present (EEPROM Shadowed in RAM)
  Software Data Prot:   TRUE
  Floppy+2S Adapter:    TRUE
  Initializing GIDE:    Master: 980x16x16       Slave: 490x16x16

20:06  SmallZ80 (H=Help) :
```

If the sign-on message appears as is should, then you're already done with the hard part – continue on with "EEPROM Setup." Otherwise, go to Chapter 6 – In Case of Trouble.

The SmallZ80 has an EEPROM Write Protection feature in the design – the /WR signal is not applied to the EEPROM unless the shorting block on J8 is set to Write Enable - pins 2-3. Write Protect is established with the shorting block on pins 1-2. In order to execute the Setup Write, you must have J8 set to WE – Write Enable. Unless you do so, any EEPROM Setup configuration you attempt to establish will fail and the system will hang.

Also, if the above display for "Software Data Prot" displays FALSE, you will be required to Write Enable the EEPROM so it can set SDP during power-on initialization. Once the display shows SDP TRUE, you can return the EEPROM to Write Protect (J8 at WP, pins 1-2). The EEPROM should be delivered with SDP=TRUE.

Chapter 2
EEPROM Setup

Enter EEPROM Setup from the command line by entering the 'S' command character. You should get the following menu:

```
SmallZ80 Setup Menu Options

 A - CPU Board Parameters
 B - Terminal Control Strings
 C - Expansion Board Parameters
 D - GIDE Parameters
 E - Set System Defaults

 F - Display All Settings

 G - ReBoot - Discard Changes
 H - ReBoot - Test Changes
 I - Reboot - Write EEPROM

Select (ESC to Exit) ?
```

Options A-E allow you to configure the System Setup in various ways. You can enter each of the options and change anything you want to see how it works, but BE CAREFUL about changing things that affect the system console configuration. Specifically, Option A – CPU Board Parameters, which looks like this:

```
Current CPU Board Parameters:
 Ser-A Baud Rate:        @ 9600     HFC: Off
 Ser-B Baud Rate:        @ 9600     HFC: Off
 Soft Reset:             Off
 User String:            SmallZ80 Example Owner String

Options:
 1 - Set Serial-A Baud Rate
 2 - Set Serial-B Baud Rate
 3 - Toggle Serial-A Flow Control
 4 - Toggle Serial-B Flow Control
 5 - Toggle Terminal Soft Reset
 6 - Set User String

Select (ESC to Exit) ?
```

Chapter 2
EEPROM Setup (cont)

Our testbed systems have been using Serial-A baud rates up to 460,800 without a problem, but if you set it and YOUR system doesn't support it, see the "Baud Rate Errors" section below. To prevent baud rate problems, you should ALWAYS adjust the baud rate, ESC back to the main Setup menu, then use the H – ReBoot – Test Changes option. The system will restart and you'll get garbage on your screen - this is normal! The Terminal is still at the old baud rate while the CPU has shifted to the new setting. DO NOT RESET THE SMALLZ80! Simply change the terminal baud rate to match the SmallZ80 setting (you may have to close the Terminal's original connection, then re-open it to shift baud rates), then press 'H' on the keyboard. If the two baud rates match, this will display the Help screen and you should be operating at the new baud rate.

If there are problems at this point, perform a hard reset on the SmallZ80 and set your terminal back to the previous baud rate (9600 for distribution configuration) to start over.

If all is operating normally, you can write the new baud rate into the EEPROM by once again entering Setup, Option A and selecting Serial-A baud rate. Verify the baud rate is as you want it, then ESC back to the main Setup menu and exit Setup via the I – ReBoot – Write EEPROM option. This will write the EEPROM Setup data and your system will always power up at the new baud rate. (Note that J8 shorting block MUST be in the WE position – pins 2-3 shorted.)

Other settings can be changed without the possibility of making the system unusable. Note Setup Option A, entry 6 – Set User String. This allows you to replace the "SmallZ80 Example Owner String" message with a 30 character string of your choice. Terminate the string with a Carriage Return. The system will prevent you from exceeding 30 characters.

Setup Option B – Terminal Control Strings, will display the current settings and allow you to enabled/disable the functions, but you cannot actually modify the strings. As delivered, the strings emulate a VT-100 terminal. A future release will allow you to modify the control strings.

Setup Option D – GIDE Parameters, is used to set the Cylinder, Head, and Sectors per Track used by the IDE Master or Slave drives. As delivered, CHS is set to Emulation @ 4096 Cylinders, 16 Heads and 16 Sectors/Track (528MB of storage.) We recommend you leave these settings as-is, although you can, if desired, set any reasonable values (16 Heads and SPT, max.) If the drive can use those settings, it will; if not, it will reject them and you'll see the result when you restart the system. The important values are Heads and Sectors/Track – the number of cylinders will be adjusted to the maximum number of cylinders commensurate with the Heads and Sectors/Track you requested.

Baud Rate Errors

It's entirely possible that you can manage to mismatch the baud rates between CPU and Terminal and not be able to get the two talking again. In this case, power down, set your terminal to 9600 baud, no HFC, and connect it to Serial-A. Install the shorting block on J2 near the UART. Power up the SmallZ80. It will connect to Serial-A at 9600 baud and Flow Control disabled, regardless of the previous Setup parameters. Be sure to run Setup, set the desired terminal parameters, and exit by writing the EEPROM. Power down, remove the shorting block on J2, power up and start over.

Chapter 2
TCP232 Setup

The TCP232 Serial Device Server from USRIOT is used to connect the SmallZ80's serial TTL output to a 10/100Base-T LAN. In this manner, LAN cabling replaces the serial cable between the SmallZ80 and the terminal. Typically, with this connection, your terminal will actually be a Windows or Linux computer running a TELNET program for terminal emulation. This combination provides the fastest possible terminal display available to a CP/M computer today.

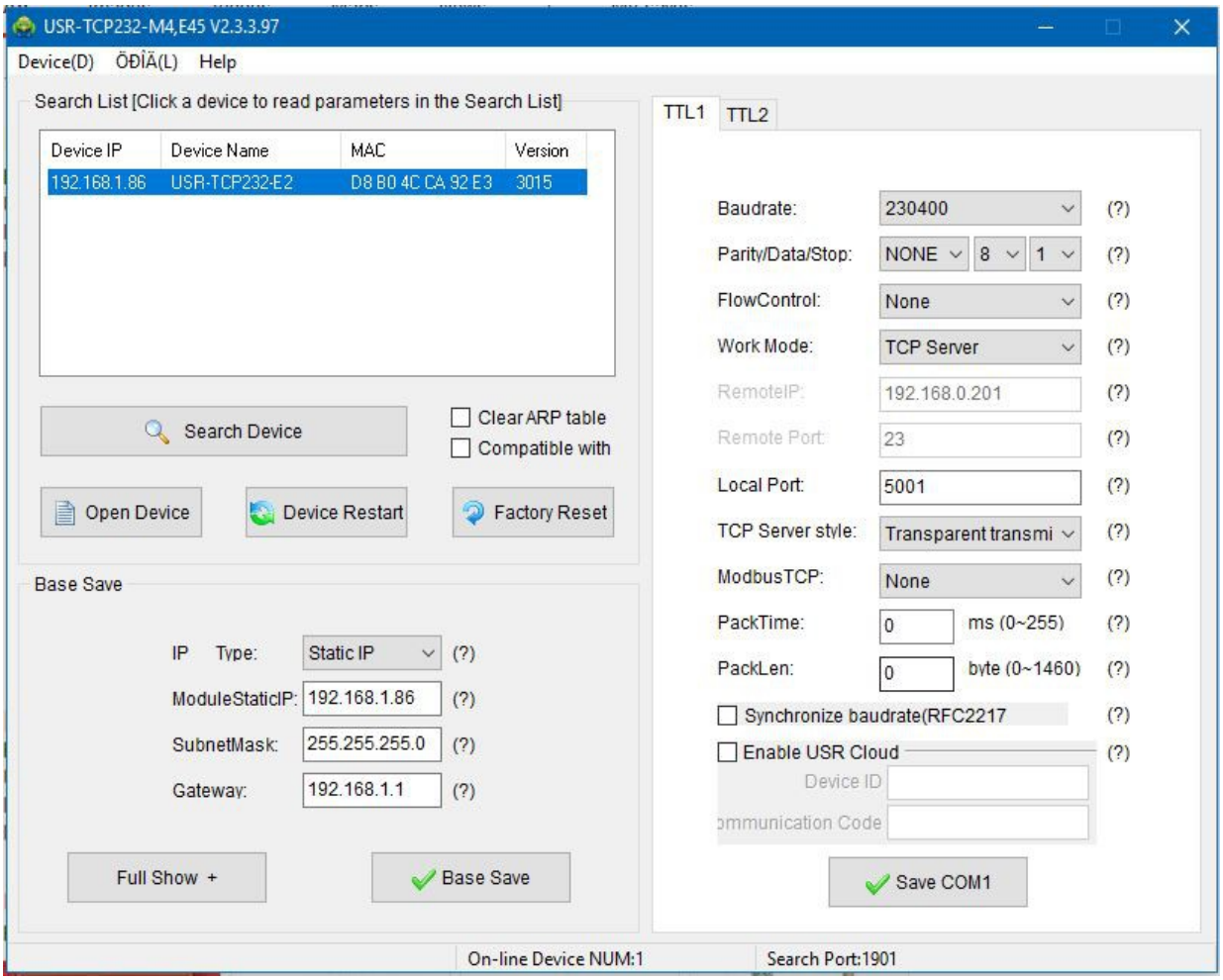
The TCP232 must be configured to work properly in the SmallZ80 environment. When we deliver the TCP232 as part of your Floppy+2S W/LAN kit, the configuration will have already been set so the system is usable, but you may want to make some changes to better fit your requirements.

For instance, we deliver the system with the TCP232 set to DHCP addressing. This means it will connect to your LAN and be supplied with an IP address by your router. The upside is that you don't have to make the connection yourself. The downside is that you have to figure out the IP address that was assigned and realize that it can change over time, making it more difficult for your TELNET program to connect to the SmallZ80.

Depending on your router, you should be able to reserve a specific IP address for your TCP232 so it will always remain at the same address. Alternately, you can set the TCP232 to Static IP addressing and assign the IP address yourself. It will always remain at that address and the router will have nothing to do with it.

Chapter 2
TCP232 Setup (cont)

Here is a screenshot of the basic TCP232 settings:



When the program starts, click the 'Search Device' button, then double click the device that matches the MAC address of your TCP232. Note the Base Save section below the Search Device box. This is where you set DHCP or Static IP addressing. When DHCP is used, the other boxes will be grayed out – they will be filled as assigned by your router. If you change anything in this box, be sure to click the 'Base Save' button to save your changes.

On the right half is the TTL side of the TCP232 – the part that connects to the SmallZ80. Set your desired baud rate – the faster the better! We've had perfect results at 460,800, and we've seen 921,600 work well, although it is beyond the UART's rating. Changing baud rates MUST be done to both the SmallZ80 EEPROM setup AND the TCP232 setup. Failure to do both will result in a failure to communicate.

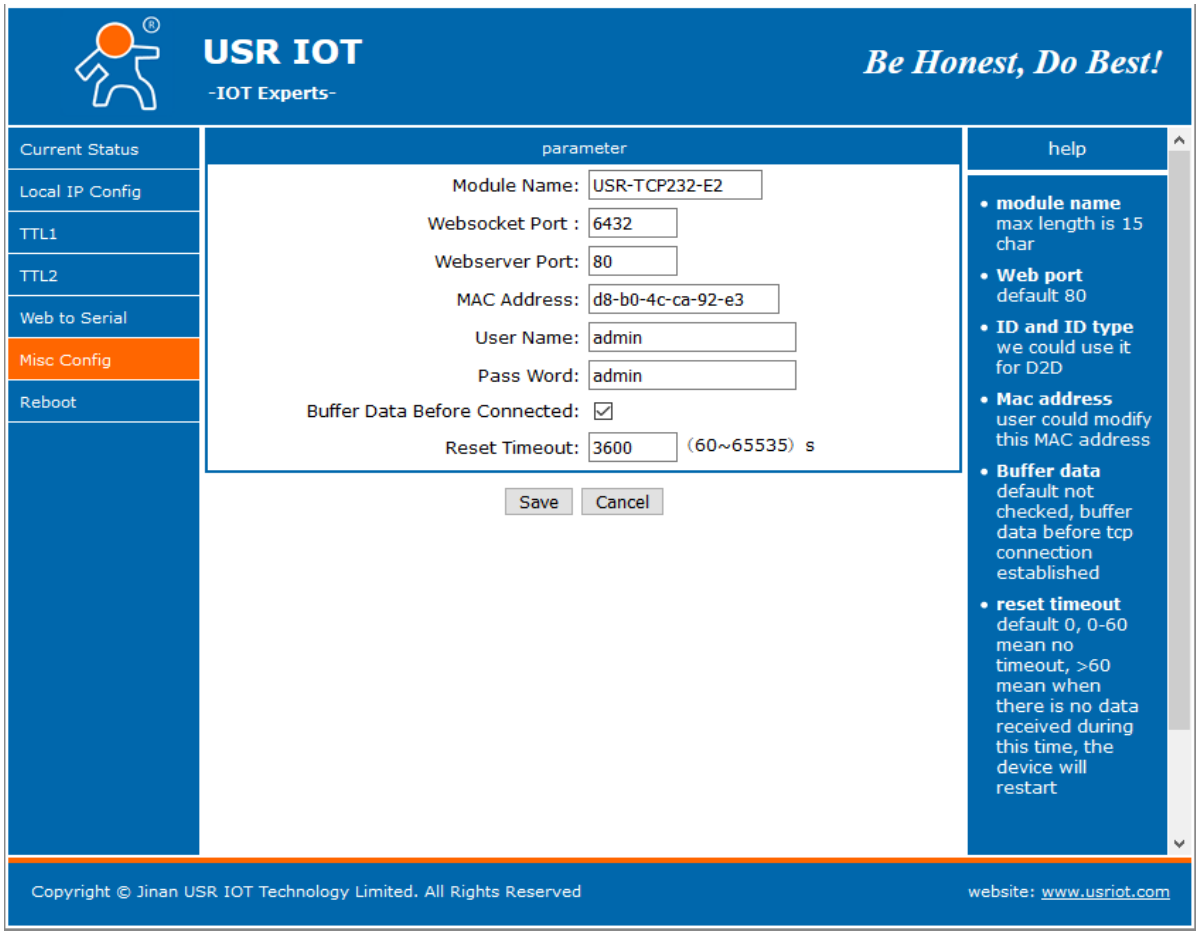
Chapter 2
TCP232 Setup (cont)

The other settings should remain as shown above. When complete, be sure to click the 'Save COM1' button.

The second serial port on the Floppy+2S expansion board can be used to drive the TTL2 port on the TCP232. Typically, this would be for a LAN-base printer, although it could also connect to a different computer or even a second terminal.

The TTL2 settings are independent of TTL1. One setting that MUST be different is the 'Local Port' setting. TTL1 is set to 5001; we use 5002 for TTL2. With the port # added to the IP address, the SmallZ80 LAN Terminal address for the diagram above is 192.168.1.86:5001. Your TELNET program will need the IP address AND port # to establish the connection.

There are two settings for the TCP232 that is not available on the setup application, but ARE available when using your browser to perform setup via HTTP. Click the "Open Device" push button on the left – it will open your browser and enter the TCP232's webserver page. Click the Misc Config box on the left and you'll get this page:



Chapter 2
TCP232 Setup (cont)

Note the “Buffer Data Before Connected” check box - it might be unchecked. If so, check it and click 'Save'. If this box is not checked, the TCP232 will discard any serial traffic coming from the SmallZ80 until after the connection to the terminal is established. This will invariably result in the SmallZ80 sign-on display being lost. After the connection is established, you can reset the SmallZ80, or press the 'R' key to restart it without a hard reset.

Or, you can simply check the “Buffer Data Before Connected” box. The TCP232 will hold any text it receives from the SmallZ80 until the terminal link is established, THEN it will send the data and you'll get a normal startup display without a reset or restart.

Also note the “Reset Timeout” setting. Any setting above 59 (minutes) will result in a Timeout condition when the TCP232 doesn't receive any data for the duration of the timeout setting. A timeout will reset the TCP232 and close the terminal connection. Disabling the Timeout with a value less than 60 will prevent the automatic TCP232 reset, leaving the terminal connected.

After setting the “Buffer Data” checkbox and timeout delay, be sure to click the “Save” button.

Chapter 3
Theory of Operation

CPU Board

CPU

A standard 40-pin DIP Z80 CPU enhanced to operate at 20MHz. See the Zilog Z80 User Manual on the SmallZ80 CDROM for particulars.

Clock

The SmallZ80 powers up with the UART Oscillator (Y2) driving SysClk at 14.7456MHz. Once RAM emulation is entered, SysClk switches over to the 20MHz oscillator (Y1.) The slower, power-on clock speed was needed because the EEPROM chips become unreliable around 20MHz. XICOR chips can run fine at 20MHz, but they do not write reliably at that speed. Atmel chips simply will not operate at 20MHz at all. Rather than run the system at 18.432MHz (which works fine for both chips) we elected to go with two CPU clocks: a slower clock for power-on and EEPROM Write, and a faster clock for normal (RAM-only) operation. The UART Channel B DTR output pin is used to control clock speed select. At power-on (or reset), DTR-B is disabled, which is a HIGH on the DTR-B pin. That HIGH level causes U18 to pass the UART clock to the SysClk clock input. Writing a '1' to the DTR bit (bit 0) of the Serial-B Modem Control Register forces the DTR-B output LOW, which toggles the high speed clock through U18 to the SysClk input, shifting the system to high speed and lighting the HiSpd LED at D5.

EEPROM

The EEPROM chip can be XICOR X28HC256P-15 or Atmel 28C256-15PU (faster is OK); it is a 32Kx8 EEPROM that can be re-programmed in-circuit provided certain protection schemes are validated. The EEPROM code uses a 256-byte 'Setup' data block that is written on demand, maintaining User settings while power is off. A CP/M utility provides the ability to write the entire contents of the EEPROM to a file, as well as reading an image file and writing it into the EEPROM. The EEPROM is placed 'in context' when GpPort bit 7 is reset. When the EEPROM is 'in context,' all memory accesses to addresses between 0000H-7FFFH are directed to the EEPROM. During power-on initialization, the contents of the EEPROM are copied into RAM, the EEPROM is shifted 'out of context,' and the program is then executed from RAM.

The EEPROM is checked during initialization to verify that the Software Data Protection (SDP) function is enabled. If not, it is enabled and a flag is set in the EEPROM. The status of SDP is displayed during initialization – it should always be TRUE. SDP will prevent any inadvertent writes to the EEPROM unless the specific unlock sequence is written first. Even then, the EEPROM is unlocked only for a very short period of time.

If SPD is displayed as FALSE during initialization, ensure the EEPROM WP/WE jumper @ J8 is set to WE (pins 2-3) and reset the system. Once the display shifts to TRUE, you can return the jumper to the WP position.

Chapter 3
Theory of Operation (Cont)

RAM

The RAM bank is comprised of two ICs: U3 (32Kx8) and U4 (512Kx8.) See Appendix E regarding the Memory Map. Depending on the Map in context, the two RAM chips are selected as decoded by the Memory Map Decoder. When the Ram-Only map is in context, addresses from 0K-16K (0000H-3FFFH) and 48K-64K (C000H-FFFFH) are directed to the 32K RAM at U3. Memory accesses from 16K-48K (4000H-BFFFH) are directed to the 512K RAM at U4. Ram Banking is used to determine which 32K 'Bank' of the 512K RAM chip will be enabled when the memory chip is accessed. GpPort bits 3-0 are used to select one of sixteen 32K banks within the 512K RAM chip. These four 'bank select' bits are tied to U4's A15, A16, A17 & A18 inputs.

General Purpose I/O Port (GpPort)

The GpPort is used to control three aspects of the SmallZ80: Memory Map Select, Post Code LEDs, and Bank Select. On power up/reset, all bits are LOW, selecting the RAM+ROM Map, all POST Code LEDs OFF, and RAM Bank 0 in context. When writing the GpPort, it is imperative that you don't inadvertently change any bits other than the ones you want to change. To do this, you are required to input from GpPort, make the required change(s), and write it back. This is because you DO NOT WANT TO change the Memory Map or the current RAM Bank, when all you really wanted to do was turn an LED on or off. This is accomplished with the following code snippet:

```
di
in    a,(GpPort)        ; Read the contents
and   10001111b         ; Turn off all 3 LEDs
or    RedLED + GrnLED   ; Turn on Red and Green LEDs
out   (GpPort),a        ; Make the change
ei
```

In this manner, the LEDs can be manipulated without changing the MemMap or the RamBank bits.

Terminal I/O

The SmallZ80 distribution code sets Serial-A as the terminal port, operating at 9600 baud, 8 bit words, 1 stop bit and no parity (8N1.) EEPROM Setup allows you to set any one of the two (or four, if the Flop+2S adapter is installed) serial ports as the terminal port. It also allows you to select the baud rate and Hardware Flow control settings. Baud rate can be set to any of the standard values from a low of 9,600 baud to a high of 460,800 baud. The system operates reliably at 460,800 baud without flow control; your terminal will determine the actual limit. The selected port is passed to CP/M at boot time, so there is no concern about changing both EEPROM and CP/M when you want to change the port, the baud rate, or the flow control setting.

If the Floppy+2S adapter is installed, the max baud rate allowed for Serial-C and -D is 921,800 baud. The standard serial port will not be able to operate that fast, but the TCP232 LAN adapter does, providing lightening fast terminal I/O.

Chapter 3
Theory of Operation (Cont)

Reset Generation

Reset is initiated at power-on or when the RESET push button is pressed (either front panel or on the CPU PCB.) The TL7705 Voltage Supervisor at U13 will immediately assert the ResetH & /Reset signals and hold them active until approx 25mSec after the 5v rail reaches 4.55v at power up, or after the reset push button is released. Once the TL7705 releases the hold, ResetH will immediately drop to a LOW state, allowing the UART to initialize. C13C is used to stretch the rise time on the /Reset signal, allowing the UART to initialize and set the HIGH level on DTR-B (establishing the CPU clock on the low speed oscillator) before the CPU begins program execution.

Memory Map Decoding

U14 is the Memory Map Decoder. It uses A13-A15, *MREQ, and /RomEn to decode the memory access to one of three chips: U2 (EEPROM), U3 (32K RAM), or U4 (512K RAM.) In addition, it uses *RD, *WR, *IORQ and *M1 to decode an I/O operation, and provides the /IoRd & /IoWr signals that are used for initiating I/O operations throughout the system. See Appendix E for a description of the Memory Map.

I/O Map Decoding

U15 is the I/O Decoder. It uses the /IoRd & /IoWr signals from U14, along with the A0-A7 signals to decode the appropriate port for a Read or Write operation. See Appendix F for a breakdown of the I/O Map.

Real Time Clock

The SmallZ80 uses an Epson 72421B Real Time Clock with battery backup. It is set via the EEPROM 'CS' (Clock Set) command. Addressed at I/O ports 50H-5FH, the RTC stores Month, Day, Year, Hour, Minute, Second, and Day of Week data. There are also three control/status register ports. The RTC can be operated in an interrupt mode, where interrupts are generated every 1/64 second, 1 second, 1 minute, or 1 hour. The SmallZ80 uses the 1/64 second interrupt.

Interrupts

The Z80 is set to Interrupt Mode 1 by the initialization code. In Mode 1, the Z80 responds to an interrupt with a RST 38H instruction, which is essentially a two-byte CALL 0038H. The interrupt handler exits via a RET instruction. The only source of interrupts on the SmallZ80 is the Real Time Clock, which is initialized to interrupt 64 times per second, thereby generating a 15.625mSec interrupt signal. The EEPROM and BIOS interrupt handlers will both ignore 3 of every 4 interrupts, responding only on the 4th interrupt. This creates a 4 x 15.625mSec = 62.5mSec interrupt handler, which comes to exactly 16 'handled' interrupts per second.

Chapter 3
Theory of Operation (cont)

POST-Code LEDs

The POST Code LEDs consist of three LEDs, usually external to the case, that can be used to track the execution of a code block. This can be especially useful for finding the location of a troublesome coding error that is causing a 'hang' and is difficult to determine the cause. The LEDs are, by convention, coded as GrnLED, YelLED, and RedLED, for their locations within the GpPort bit mask (bit 6=GrnLED, Bit 5=YelLED, & Bit 4=RedLED.) There are a handful of LED operations used by the EEPROM code that allow you to track the progress of various operations, tests, and procedures.

Floppy + 2S Adapter

Floppy

The SmallZ80 floppy port can drive two standard 3.5” 1.44MB floppy drives with a 'twisted' cable. Unlike the common PC configuration, the drive BEFORE the twist is Floppy 0:; the drive AFTER the twist is Floppy 1:. If needed, there is a 5vdc header (J11) on the main PCB that can supply +5vdc for 1 or 2 floppy drives. Solder the red/black wires directly to the PCB; red goes in the square pad (pin 1.)

Serial Ports

The two Flop+2S serial ports (Serial-C and Serial-D) are standard, 9-pin DTE serial ports (the same configuration as a standard PC serial port.) If present, Serial-C is tied to the CP/M Reader device; Serial-D is tied to the Punch device.

EEPROM Setup is used to set the baud rate and Hardware Flow Control parameters for both ports. These serial ports are allowed to run up to 921,600 baud. Although the serial connections do not operate properly at that speed, the LAN connections ARE reliable at 921,800 baud, and will provide a lightening fast terminal display.

If desired, you can install the TCP232-E2 Ethernet LAN addition on the Flop+2S adapter, replacing the two external serial ports with LAN ports. There are 3 shorting blocks on the adapter that are positioned to either 'S' (for Serial) or 'L' (for LAN). There are no specific settings in Setup that apply – as far as the machine can tell, the LAN adapter is simply two serial ports. When the LAN is used for Terminal I/O, Serial-D is the LAN Terminal port and Serial-C is the LAN printer port. Serial-A becomes the CP/M Reader device and Serial-B becomes the CP/M Punch device.

The three shorting blocks are used to determine if the Serial inputs to the UART come from the LAN adapter or from the serial port level shifters. There is no similar selection process for the UART outputs – they go to BOTH destinations (LAN and level shifters) at all times.

Chapter 3
Theory of Operation (cont)

Ethernet LAN Adapter

The LAN adapter module is a TCP232-E daughterboard. It uses PC-based software (on the CDROM) to configure it for use with the SmallZ80. The Installation/Setup section provides specifics. A second method of configuration is via HTTP and your Internet browser. Simply enter the IP address of your TCP232-E module on your browser's address line.

When the LAN adapter is installed, the SmallZ80 can run terminal I/O at 460,800 baud and flow control disabled with no loss of characters. Screen updates are extremely fast. Testing at higher baud rates continues, but we've had excellent results at the max speed of 921,800 baud.

The conversion from Serial I/O to LAN-based I/O is invisible to the user and the SmallZ80 – the TCP232 handles it all. To the SmallZ80, LAN I/O is nothing more than serial I/O over a 16C550-based serial port.

There are no concerns about DCE vs DTE ports, male vs female or 9-pin vs 25-pin connectors. It's all handled with a single RJ-45 connected to your LAN. Baud rate issues still remain – the SmallZ80 and the TCP232 must be in sync on the baud rate. Typically you would use the SmallZ80 setup program to set the baud rate and exit EEPROM setup via the 'Write EEPROM' option. Then, via HTTP or the USRIOT TCP232 Setup utility, you change the TTL1 baud rate, save the COM1 settings, then restart the TCP232 by clicking the 'Device Restart' button.

Once the TCP232 restarts (a couple of seconds), you will then have to re-establish the connection on your TELNET program. When the TCP232 is restarted to establish the new baud rate, it 'broke' the previous TCP link to your TELNET terminal, so you'll have to restart it. The method by which you do so varies with the software. On Absolute TELNET, you click the red 'X' to close the previous connection, then click the green check to re-open it.

Chapter 4
Operating System

Z-System Basics

The Z-System is a module-based replacement for the stock CP/M 2.2 from Digital Research. The CCP and BDOS are replaced with ZCPR3.4 and ZSDOS, while the BIOS is modified to provide the initial system configuration during cold boot. When cold boot is complete, ZCPR executes a batch file that loads the various system segments into RAM. These segments vastly increase the capabilities of the Console Command Processor (CCP) while still being limited to the standard CCP size of 2048 bytes.

The segments provide several enhancements to the base CP/M”

- The Resident Command Package (RCP). Nearly all of the CCP-based commands are offloaded from the CCP and placed in the RCP, providing space for many enhancements to the CCP itself.

- The Flow Command Package (FCP). Flow control of batch file processing.

- Named Directory table (NDR). In addition to Drive and User (DU:) numbers (such as A0:) you can assign a name to a DU: and refer to the name rather than the DU: format.

- Environment (ENV). A small table that describes the system segments, their size and locations, and a wealth of other information to any program that knows to look for it.

- Z3 Terminal File (Z3T). Describes the capabilities of the terminal and how to execute several of the control strings the terminal responds to, such as cursor movement, clear screen, graphics, reverse/normal video, etc.

SmallZ80 OS Intro

On the SmallZ80, the RCP, FCP, and NDR are in 'banked' memory, using far less TPA space than would otherwise be required. This also allows them to be expanded significantly, providing far more options to the user.

SmallZ80OS Build Options

The various build options are specified in file A3:BIOSZ.Z80. Here you can select the drive types as well as the various BIOS optional parameters.

Generating a Boot Drive or Disk

The process of generating a boot disk/drive is the same regardless of the destination being a Compact Flash partition or a floppy drive. From the A3: User Area, simply enter the command 'MAKEIDE x', where 'x' is the destination drive, such as MAKEIDE A (do not append the colon character to the drive letter.) The ZEX batch command processor will take over and compile the

Chapter 4
Operating System (cont.)

BIOS, relocate the code to the established address, and write the resulting CCP, BDOS, and BIOS to the specified drive or partition. You must also copy all files from User Areas A0:, A4:, A14:, A15:, and A16: to the destination. The batch file alias MAKEFLOP.COM on A0: will copy all required files to the K: floppy drive.

Floppy disks should be formatted prior to use. The FORMAT.COM program has been configured to use the optimum skew for the SmallZ80, but any skew from 1-9 is allowed. A stock PC-formatted disk will work for read/write, but it will be somewhat slower than a disk formatted for the SmallZ80.

Building Data Drives

Any floppy or IDE partition can be used as a 'data drive.' The name 'Data Disk' implies that the disk is not a boot disk, but there is no physical or logical difference between the two. Both have boot tracks set aside for operating system; both have the same amount of storage and the same directory size.

CP/M Utilities for the SmallZ80

FORMAT.COM – Used to format floppy disks. Defaults are Floppy 0:, Interleave=6.

Options:	//	Display Help screen
	/Dx	where x = 0 or 1 for desired floppy Drive
	/Ix	where x=1-9 for Interleave
Usage:	FORMAT [/Dx] [/Ix]	

HDPART.COM – Used to display/change the current partition assignments.

Options:	//	Display Help screenshot
	Cx	Set drive C: to partition x (0-9)
	?	Display current partition assignments
	R	Reset all partition assignments to default
Usage:	HDPART c7,b8,d3	Assign C: to partition 7, B: to part 8, D: to part 3
	HDPART R	Reset all partitions to defaults

ZED.COM – Line oriented text editor, with source code color context.
Enter HELP from command line, then select B – Z-EDIT for details.

ZZAP.COM – Disk/File sector editor.
Prompts on screen. No specific instructions.

Chapter 5
Programming Considerations

General Purpose Port

The General Purpose Port (GpPort) controls three functions: 1) Memory Map, 2) POST LEDs, and (3 RAM Bank Select. When managing one of the functions, it is imperative that care be taken not to mistakenly change one of the other functions.

The GpPort is accessed and modified by the interrupt handler, so it is advisable to disable interrupts before beginning any GpPort modification.

Bit 7, when SET, enables the RAM-Only memory map. When Reset, the RAM+ROM map is in effect.

Bit 6, 5, & 4, when set, light the Green, Yellow, & Red LEDs (respectively.) Resetting the appropriate bit(s) will turn the LED(s) off.

Bits 3, 2, 1, & 0 determine which of the sixteen 32K RAM Banks is in context. When all bits are 0, Bank0 is in context; when all bits are 1, Bank15 is in context.

At Power on/Reset, the contents of the the GpPort register are 00000000b.

Real Time Clock

The Real Time Clock (RTC) occupies the port address range of 50H-5FH. The RTC is programmed to generate an interrupt 64 times per second (15.625mSec), but only one of every four interrupts is actually processed, giving the system a 62.5mSec 'heartbeat' interrupt. Thus, there are 16 'handled' RTC interrupts every second.

The EEPROM and BIOS code read the RTC date/time routine every 16 seconds. When the EEPROM is in control, the HH:MM time is updated on the terminal command line every time the minute changes. Under CP/M, the WORD ADDRESS of the current Date/Time string is stored at location (0027H.) For instance, 0027H-0028H has the word F3B6H, so the 18-byte date/time string starts at F3B6H in the format of: '10/18/18 14:21:32' (the quote characters are not in the buffer – they are used here only for clarity.)

Chapter 5
Programming Considerations (cont)

External BIOS

In order to minimize the BIOS footprint as much as possible, a number of routines have been offloaded from the BIOS into Banked memory – this is referred to as the External BIOS. At this writing, the External Bios only uses one of the 32K RamBanks, and only a very small portion of it. Page0 carries several pointers into the External BIOS:

0029h-002AH	Address of ExtBios Buffer
002BH	NZ = ExtBios loaded
002CH	JP instruction
002DH-002EH	ExtBios entry address

Calling conventions:

```
Enter with: 'C' holds Service Function
            'B' holds Service Subfunction (if any)
            'A' has single-byte data
            'HL' has 2-byte or word data, or address pointer

Rtn with:   'C' set if error w/error code in 'A'
            'HL' has address of ExtBIOS data buffer in BIOS memory

Error Codes returned in 'A':
      A=1     Out of Range; Function or SubFunction not recognized
      A=2     ExtBios Not Loaded
```

Using the above information, you can easily determine if the External Bios is in place by testing the byte at 002BH for Z or NZ. If NZ, the External Bios is loaded and ready. Then, simply CALL 002CH to enter the External Bios. Be sure to load the A, B, C, and HL registers as required by the calling function as described in file A4:EXTBIOS.LIB.

On return, HL has the address of the External Bios Buffer for any returned data. The Buffer address word is also in Page0 at location 0029H-002AH.

Chapter 6
In Case of Trouble

This sections needs a lot of work, obviously. Contact TerryG@stack180.com for assistance.

General Problems

Power Supply Problems

Dead System

Interrupt Problems

The interrupt system can be tested via the EEPROM Testing command, Option B – Test Heartbeat 62mSec Interrupt. Once you select Option B, the test will begin. 2 seconds later, the test completes and displays the Press any Key message. If the machine hangs and never displays the message, then the RTC is not interrupting the CPU as it should. You should re-initialize the RTC. Turn off power and remove the battery jumper at J7. Wait 30 seconds, then re-install the jumper. Power up. Enter the CS command (Clock Set) and set the time/date. Enter the Testing menu and execute Option B again. If the problem persists, contact TerryG@stack180.com for assistance.

Terminal Problems

Configuration Setup Problems

Garbled or Missing Sign-on Message

As stated above, the most common problem with a garbled message is incorrect baud rates. The SmallZ80 EEPROM is delivered set to 9600 baud, 8N1, no handshaking. Be sure your terminal uses the same settings.

Check to be certain you didn't mix up the two oscillators. The CPU oscillator is 20.0MHz (or 24MHz,) installed in the Y1 location below the CPU. The UART baud rate oscillator is 14.7456MHz, installed in Y2, just below the PLCC-44 UART at U5.

If you find that the terminal baud rate is set properly and you cannot establish communications, contact us via e-mail at terryg@stack180.com for further assistance.

For a missing sign-on message, there are several possibilities:

- Is the CPU actually RUNNING? Verify the red HALT LED (D1) is off. If you have a logic probe, verify the address and data lines are cycling. Check the U15 GAL, pin 18, to verify activity on the Serial-A select signal. This pin should be constantly cycling hi-low-high as the CPU strobes the serial port for terminal input.

- Be sure the terminal is connected to Serial-A instead of Serial-B. Serial-A is the J6 “D”

connector nearest the 4-pin power connector at J1.

- Be sure you're using a NULL MODEM connection. A DTE port cannot be connected to a DTE port; only DTE-DCE will work. A NULL modem cable or connector will make the required wiring swaps.

If all else fails, don't hesitate to contact us via e-mail at terryg@stack180.com.

Chapter 8
CDROM Contents

CDRomRoot:\Software Files Subdirectory

Absolute TELNET Subdirectory
Contains the Absolute TELNET (AT) 9.18 installation file. AT is used as the primary TELNET program for the LAN-based terminal. Hosted at www.celestialsoftware.net, AT is regularly updated and available at a good price. The current version is v9.18, dated 7 Jan 2011, and is available for download at www.celestialsoftware.net/download-selection.html. Demo, Lite, and Professional versions are all supported by the same download. The two .TNT files are configuration files used by Absolute TELNET to configure itself for the SmallZ80. They will have to be modified for your installation. They redefine certain keys on the PC keyboard to provide the key sequences used by WordStar. For instance, PgUp, PgDn, Home, End, arrow keys, etc, are set to generate the required ESC sequences that WordStar uses for those functions. The same WordStar arrow key definitions are also used in the Z3CPR TCAP definition.

Model TCP232-E Subdirectory
Contains the published datasheet and hardware manual for the LAN adapter. Also includes the M4K3 Setup Software v2.3.3.97.exe installation executable.

Tera Term Subdirectory
Contains v3.13 of TeraTerm.

ZCPR3 Subdirectory
Contains .PDF versions of the ZCPR 3.3 User Manual, Installation Manual, and the Installation Workbook.

ZSDOS Subdirectory
Contains ZSDOS User Manual in TXT format.

Boot ROM Subdirectory
As the name implies, this is the EEPROM Source Code. After installing ZDS 3.68, double-click on the SmallZ80 RevE BootRom.ZWS file to open the project. You'll almost certainly get an error saying the directories have been moved – allow ZDS to correct the problem and re-direct the pointers to the correct locations.

Win32 Disk Imager.exe
Installation file for the Win32 Disk Imager utility program that can read/write CF modules. Used after downloading the SmallZ80 CF image, it will write the new image to your CF module.

Zilog ZDS368.exe
This is the Integrated Development System for the SmallZ80 EPEROM source code. Install ZDS368 on a Windows computer. It will run under Win7/8/10, with caveats. Do not open ZDS, then use the File|Open command – the program will abort and exit. Instead, double clock on the SOURCE CODE file or the .ZWS project file.

Software Files also includes various data sheets for some of the hardware used in the SmallZ80.

CDRomRoot:\Disk Images Subdirectory

Contains the most recent SmallZ80 CF image from the Win32 Disk Imager program.

CDRomRoot:\Photos Subdirectory

Contains some photos of various PCB-Case installations.

CDRomRoot:\Walnut Creek Subdirectory

Contains the entire contents of the Walnut Creek CP/M CDROM.

Appendix A
EEPROM POST Codes

POST codes are used as a way for the machine to keep the User apprised of the current status of the machine. For instance, Memory test loops can blink the LEDs, Boot code can cycle the LEDs through a progressive sequence. When something goes wrong, the LEDs can point the User to the last known good code, making it easier to localize the errant code.

There are several places in the EEPROM where the LEDs are used. Most notably, the Boot code, from either an IDE or Floppy device, will display codes that identify the operation about to begin.

The LEDs are arranged in a row (or column) of three. Each LED corresponds to a bit in the GpPort register, where the leftmost bit (or topmost) (identified as GrnLED) is in GpPort Bit 6. YelLED is GpPort Bit 5 and RedLED is GpPort Bit 4. Together, these bits form a 3-bit binary code, ranging from 0-7 decimal.

Initialization POST Codes

Code	Meaning
111	RAM Present, Initializing
010	ROM Shadow to RAM Complete, RAM-based program in control
100	Hardware Init Complete

The init process continues with Terminal I/O. Ram banks are counted, EEPROM Software Data Protection checked, GIDE CHS displayed.

000 (LEDs are all off) Init is final, waiting for command input

Boot Process POST codes

Code	Meaning
001	Starting Boot Sector Read (Track 0, Sector 1)
010	Validating Boot Sector
011	Reading OS sectors
100	Transferring program control to BIOS

Other code areas cause the LEDS to flash or increment sequentially, simply as a method to show activity. Examples include memory tests and IDE access for copy or wipe routines.

Note about EEPROM Software Data Protection

EEPROMs have what's called Software Data Protection (SDP) that is used to prevent unwanted access/modification of the EEPROM's contents, such as during power on/off cycles when the system is unstable. When SDP is enabled, it is impossible to affect the EEPROM contents without first issuing the SDP unlock sequence. The SDP unlock sequence is a series of specific bytes written to specific memory addresses. Once the EEPROM is unlocked, it can be written. The number of bytes that can be written varies with the manufacturer. For the Atmel 28C256, a page write maxes out at 64 bytes; the XICOR page write maxes out at 128 bytes. Approximately 5mSec after the last write, the EEPROM will return to the locked state without further action. Writing the next page of data requires that the EEPROM be unlocked again.

Appendix B
EEPROM Parameter Passing

As the EEPROM code prepares to pass program control over to the Cold Boot BIOS, it initializes or transfers certain data to RAM Page 0 that is used by the BIOS to determine certain system operating fundamentals that were set by the EEPROM. The data consists of:

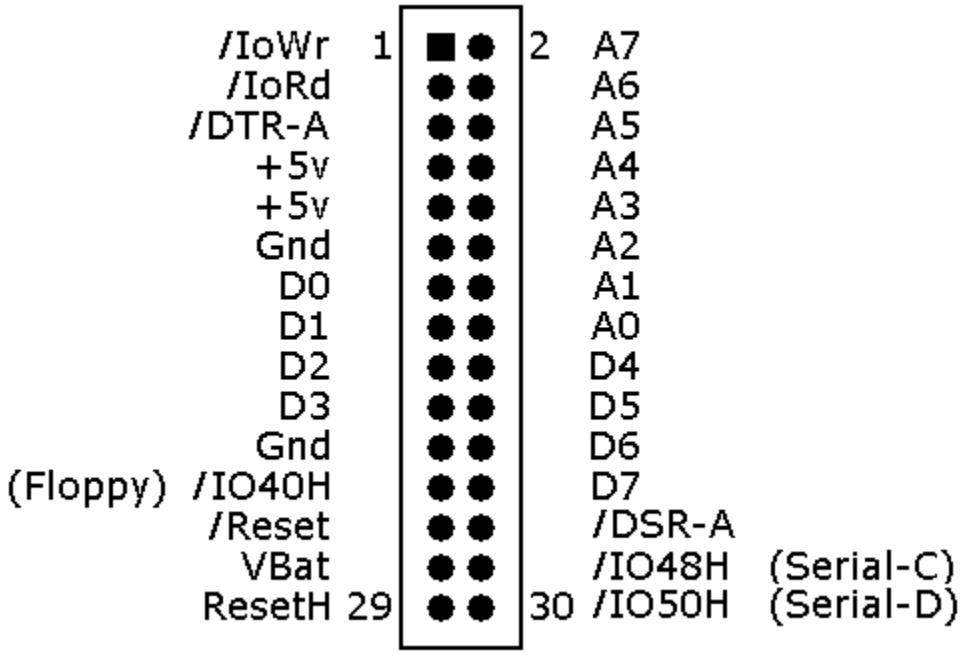
```
Address     Contents
0000h       JP to BIOS Warm Boot address
0004H       Disk Select (Drive A = 00H)
0005H       JP to BDOS entry point
0013h       BootDpe - Boot Device identifier
0014H       Boot Device Select (Master or Slave)

0040H       BIOS Data Table Base
0040H       Console Port Base address
0041H       Console Port HFC setting
0042H       Serial-A Base port
0043H       Serial-A HFC
0044H       Serial-B Base port
0045H       Serial-B HFC
0046H       Serial-C Base port            (NULL (False) if not present)
0047H       Serial-C HFC                  (NULL (False) if not present)
0048H       Serial-D Base port            (NULL (False) if not present)
0049H       Serial-D HFC                  (NULL (False) if not present)
004AH       # of Main RAM Banks (RamBanks)
004BH       Expansion Board Type flag
            Type Flag value:  0 if no Exp adapter installed
                              1 if Flop2S installed
                              2 if Lan+ALFAT installed
                              3 Not Assigned
                              4 Not Assigned
004CH       ALFAT Initialized flag        (NULL if not initialized)
004DH       ReRoute Port Base address
004EH       ReRoute Enable Flag           (NZ=Enabled)
```

Most of the bytes are obvious settings from EEPROM Setup, such as the Hardware Flow Control settings for each of the serial ports. The BIOS Data Table is used to pass a table of data directly to the BIOS, but it does not take up Page0 space after Cold Boot is complete. The table is copied into BIOS space and a two-byte pointer (word address) is stored in Page0 to provide access to the table from outside the BIOS. The table pointer is defined in the Page0.LIB file as BDatTblAdd EQU 0026h. Page0.LIB also provides offset equates into the table elements.

Appendix C
Expansion Connector

The SmallZ80 has a single expansion connector, placed roughly in the middle of the PCB. It passes all of the signals necessary to generate any type of I/O expansion board.

Arranged in a 2x15 format, the expansion connector is intended to be a female socket header that mates to a male header on the bottom of the expansion board. However, a 30-pin ribbon header (male) can be used instead. In this manner, a ribbon cable can be used to extend the connection to a remote PCB mounted elsewhere.



The three I/O signals (/IO40H, /IO48H, & /IO50H) are each 8-port I/O ranges. For instance, the /IO40H signal is active low on any I/O access from ports 40H-47H. Likewise, /IO48H is active in the range of 48H-4FH and /IO50H is active in the range of 50H-57H. When used with the current Floppy + 2S adapter, the ports supply select signals for the Floppy, Serial-C, and Serial-D devices.

/DTR-A is a single-bit output signal available for any use, and can be read or changed by accessing the Serial-A Modem Control Register, Bit 0. /DSR-A is a single-bit input signal, also available for any use, and can be read by accessing the Serial-A Modem Status Register, Bit 5.

For expansion boards mounted to the main PCB, there are 4 support standoff locations that can be used with 3x6mm machine screws and 12mm F-F standoff posts.

Appendix D
Memory Map

The SmallZ80 uses two memory maps, depending on whether or not the EEPROM is in context. This is only the case immediately after Power On or Reset, after which the RAM-Only memory map is in effect.

EEPROM+RAM Memory Map

The Power On/Reset (ROM+RAM) memory map consists of the 32K EEPROM and two 16K banks of RAM, for a total of 64K. The 16K upper half of Banks 0-15 can be selectively placed in context from 8000H-BFFFH (32K-48K.) The 16K lower half of the Banked RAM area is actually the upper half of the EEPROM code, so Bank switching does not affect the 16K-32K RAM area.

Ram-Only Memory Map

The memory map changes during the power-on initialization process, whereby the contents of the EEPROM are copied into the Lower half of the 32K RAM and the lower half of RamBank0. At that point, the EEPROM is switched out of context and the RAM Only map is in effect.

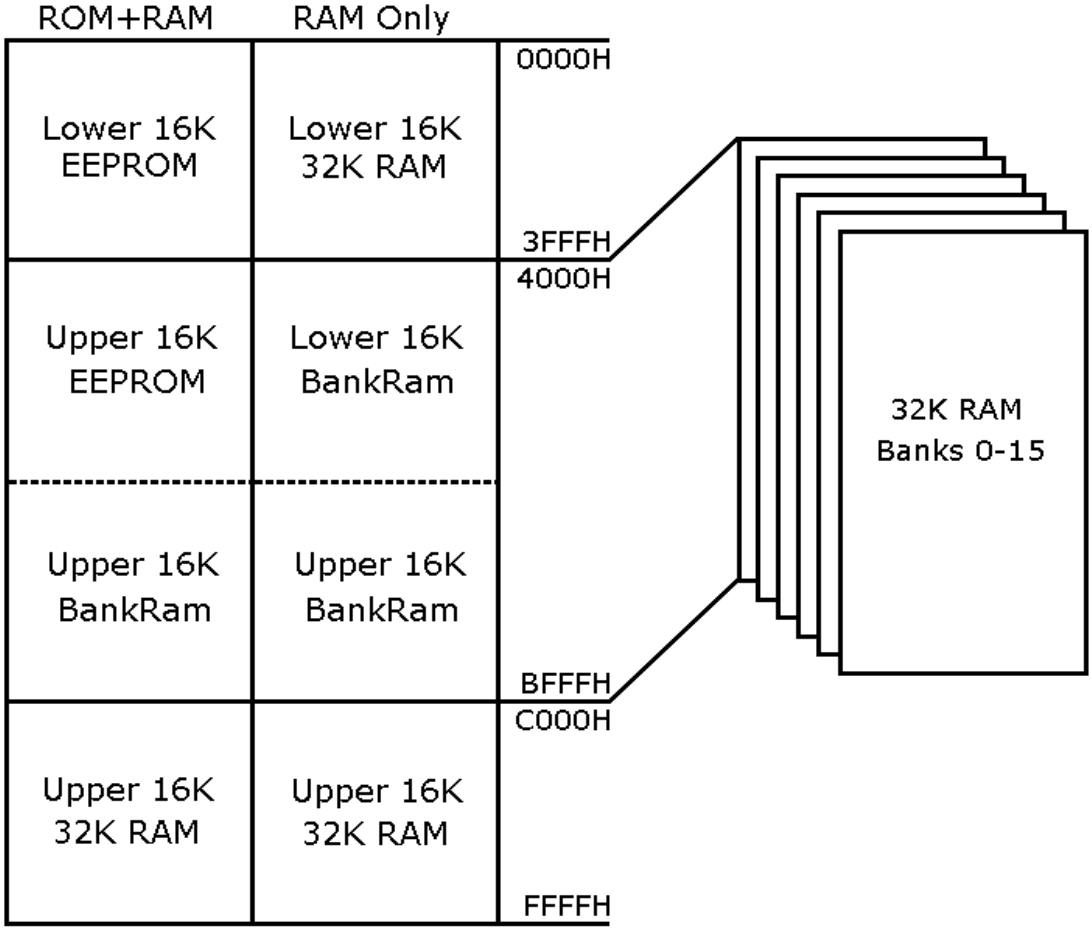


Figure D1: Memory Map

Appendix E
I/O Map

The I/O subsystem is driven by U15 – a GAL16V8-15. It performs a sum of the various signals to decode the active I/O addresses used throughout the SmallZ80. The map has been modified to move the ports above 3FH for the Z8S180 CPU.

```
Address           Usage
0000H-003FH       Reserved for Z8S180 CPU adapter
0040H-0047H       Serial-A Ports
0048H-004FH       Serial-B Ports
0050H-005FH       Real Time Clock
0060H-006FH       GIDE
0070H-0077H       Floppy Ports (Exp Connector)
0078H-007FH       Serial-C (Exp Connector)
0080H-0087H       Serial-D (Exp Connector)

00F8H             General Purpose (GpPort) Control Port

GpPort is a bit-mapped port that controls three functions:
1xxx xxxxB  Bit 7 = 0 for RAM+ROM Memory Map (power on/reset)
                    1 for RAM-Only Memory Map (all other times)

x111 xxxxB  Bits 6 (Grn,) 5 (Yel,) & 4 (Red) control the front panel POST LEDs
                  = 0 = LED is off
                  = 1 = LED is on

xxxx 1111B  Bits 3, 2, 1, & 0 are used to select one of sixteen 32K RAM banks.
```

Notes:

- Care must be taken when manipulating Memory Map Select (bit 7). The change is instantaneous and MUST be made from a memory location ABOVE 7FFFH. If not, the code that made the change will be taken out of context and 'disappear' from the active memory map.

- The LED bits can be turned on or off in any combination. Be certain not to disturb non-LED bits as they directly control the memory subsystem. Additionally, the port is being accessed under interrupts, so it would be a good idea to disable interrupts prior to accessing GpPort. Re-enable interrupts when done. The correct way to change/control the LEDs is to read the port, mask out the original LED bits, OR in the new bits, then write the port:

```
      di
      in    a,(GpPort)        ; Read the port contents
      and   10001111b         ; Mask out the old LED bits
      or    0xxx0000b         ; OR in the new LED bits
      out   (GpPort),a
      ei
```

- Similar to the Memory Map bit, care must be taken to ensure the Bank bits are ONLY manipulated from a memory area OUTSIDE of the 16K-48K (4000H-BFFFH) Bank area.

Appendix F
Terminal Serial Connections

The SmallZ80 Terminal port is set to Serial-A by default, but can be set to Serial-B, or, if the Flop+2S adapter is installed, Serial-C or Serial-D, by the EEPROM Setup routine.

In all cases, the serial ports are configured as DTE ports and will properly connect ONLY to a port configured for DCE (which is liable to only be a MODEM.) To get around this configuration issue, the use of a NULL Modem connection cable allows you to connect the SmallZ80 DTE port to a DTE port on the Terminal (or computer-based terminal software.)

The SmallZ80 is delivered to run at 9600 baud, 8 bit word, 1 stop bit and no parity (8N1.) This is the SLOWEST baud rate available. Virtually ALL terminals from the 70s to today can run 9600 8N1. However, this provides a pitifully slow terminal display. Some of the old dedicated terminals can run up to 19,200 baud, but they may have problems with character overrun, where the next character is received before the terminal has processed the previous character. At best, this causes missed characters. More often, it causes garbled AND missed characters.

To prevent overrun, the SmallZ80 supports RTS/CTS hardware flow control. When the Host and the terminal use the same methods of flow control, overrun and garbled characters don't happen because the transmission is 'throttled' to only go as fast as the receiver can accept characters.

Faster transmissions means faster display updates. There are two simple methods of increasing the transmission rate: use of USB-to-Serial NULL Modem cables, or using an Ethernet LAN adapter for Terminal I/O. Both methods work well with the SmallZ80 and allow for much higher baud rates. A USB-to-Serial cable, in conjunction with a decent performance Windows/Linux box, can easily support 230,400 baud without any form of handshaking.

Even higher baud rates have tested well. 460,800 baud is proven on both USB-to-Serial and the LAN ports. 921,600 is available on the Flop+2S ports, primarily due to the TCP232 LAN adapter which is completely reliable at that speed. Standard RS-232 serial ports exhibit unstable characteristics above 460,800 baud – we recommend you only use 921K for the LAN connection.

If you have a local Ethernet network, you can use the LAN adapter portion of the Flop+2S adapter board. It converts the two serial ports to Ethernet ports; one of them would normally be assigned as the Terminal (Serial-D), the other to a LAN-based printer, perhaps. In this manner, terminal and printer connections are part of the LAN and there are no concerns with DCE, DTE, serial cables and mating connectors, etc.

LAN-based printers do work well with the SmallZ80, but care should be taken in choosing the right printer. It's pretty much a given that InkJet, BubbleJet and the like WILL NOT WORK. These types of printers rely on the computer driver software to create a bit-mapped image of a page that is then dispatched to the printer. They simply don't understand the ASCII code used by text-based CP/M machines. Two printers that are known to work well are the HP LaserJet 4P (with an Ethernet-to-Parallel converter)(circa early 90's), and the current model HP LaserJet PRO M400 series (specifically, the M401n model.)

If you find a printer that works, let us know and we'll add it to the list.

Appendix G
Updating the BootRom

The BootRom code is being updated regularly and provided on the Stack180.com website for download. If you have an EEPROM burner, burning the downloaded .HEX file is the easiest, fastest, and most certain method of updating the EEPROM. It is always safest to use a second EEPROM for burning the latest code, even when you use a burner.

Two alternative methods are also available. The better of the two is to install an alternate EEPROM in place of the 32K RAM chip and burn it in place. This method has the benefit of protecting the current EEPROM. If a second EEPROM is not available, then burning the Main EEPROM in place is the only alternative. Bear in mind, a mistake or unexpected anomaly can easily render the SmallZ80 unable to operate. Using a second EEPROM is preferable.

There are several steps to take in preparation, followed by several steps to the process.

Preparation

1. If you have a burner, simply download (or receive via e-mail) the EEPROM.HEX file and follow the normal process for burning an EEPROM chip.

If you intend to use the in-circuit software burner you MUST already have an EEPROM with v2.00 or higher installed in the SmallZ80 – the software burner code is not present in earlier EEPROM versions. The source for the new EEPROM data file will be from one of two places: either the downloaded CF image or a downloaded (or e-mailed) Binary file for use on a USB Thumb drive with the LAN+ALFAT adapter. The two processes are nearly identical. The CF process will be described first, then the differences when using a USB Thumb drive will be detailed.

2. The CF image can be downloaded from the Stack180.com website. Any image posted after 1 January 2019 will include the EEPROM binary in an otherwise unused area of the image. Download the CF image and use Win32DiskImager to write it to your CF module (minimum 128MB.) Doing so will completely overwrite anything on your CF card, so be certain you have backups for your data.

Install the newly-written CF module as the Master IDE drive in your SmallZ80. Install an Alternate EEPROM in place of the 32K RAM chip and place a shorting block on J15.

If you intend to burn the Main EEPROM rather than the Alternate, you have chosen to accept the consequences should something go wrong. In this case, install the CF module as described, leave the 32K RAM chip in place, and install the jumper on J15.

3. Power up the SmallZ80. It will follow the normal boot sequence right up until you would expect the command line prompt. Instead, it will execute the EEPROM Copy/Burn code. Additionally, the system will remain in Slow speed, interrupts will be disabled, and the EEPROM code will be executed directly from the EEPROM rather than being shadowed into RAM.

Appendix G
Updating the BootRom (cont)

4. The Burn Menu will look something like this:

```
 EEPROM Copy/Burn Menu

 (Alfat options not available)

  A - Copy Current EEPROM to IDE Master
  B - Copy Current EEPROM to ALFAT U0:\SmallZ80\EEPROM.BIN

  C - Burn Alternate EEPROM @ 32KRAM from Current EEPROM
  D - Burn Alternate EEPROM @ 32KRAM from IDE Master
  E - Burn Alternate EEPROM @ 32KRAM from ALFAT U0:\SmallZ80\EEPROM.BIN

  F - Burn Current EEPROM from IDE Master
  G - Burn Current EEPROM from ALFAT U0:\SmallZ80\EEPROM.BIN

Select (ESC to Exit) ?
```

(The “Alfat options not available” line will not be displayed if an ALFAT USB module is installed in the system.)

The EEPROM file is already present on the CF, so selecting option A will overwrite the new file with the current (out of date!) EEPROM data. Don't do that! Option D will read the file from the CF module and burn it to the EEPROM installed in the 32K RAM socket.

If you're burning to the Main EEPROM, select option F instead.

You will be prompted to press return to begin or ESC to exit. Once you press return, the process is fairly quick and fully automatic. Once the burn is complete, the system will return to the Burn menu. Shut down power.

If you burned the Alternate EEPROM, then remove the Main EEPROM and set aside. Remove the Alternate EEPROM from the 32K RAM socket and install it in the Main EEPROM socket. Install the 32K RAM chip back into the 32K RAM socket. Remove the jumper from J15.

If you burned the Main EEPROM... Watch the system. If it returns to the Burn Menu, then you probably got a good burn. If the HALT light is lit, or the system seems to be 'hung', press RESET and watch it re-enter the Burn menu. Select option F to burn the EEPROM again. When complete, Power up. You should see the normal startup into the command line. The sign-on display will show the new version/date information; the update is complete.

The EEPROM image is delivered in one of three ways: via the HEX file for direct use by your EEPROM burner, or either a CF image file or an ALFAT binary file for in-circuit burning via the EEPROM software (requires v2.00 EEPROM.)

Appendix G
Updating the BootRom (cont)

Download the latest CF image from Stack180.com and write it to a CF module. If you use the module currently installed in the SmallZ80, everything on the CF will be overwritten, so be certain that you have backups of everything you put on the CF module. Use the Win32DiskImager program to read the image and write it to the CF module. Install the module in your SmallZ80 as the Master device.

2. It is recommended that you use a separate EEPROM for the update. This will protect the current EEPROM from any chance of disabling the entire system. If you choose to gamble with the active EEPROM, you will only be able to recover from an error by using an EEPROM burner. This might require you to send the EEPROM to Terry @ stack180.com for re-burning.

To make matters easier, you can purchase the EEPROM Burn kit. It includes a second EEPROM, a replacement for C8A, and a ZIF socket for your SmallZ80.

If you choose to continue without the extra EEPROM and ZIF socket, skip down to step 5.

3. Place the J8 shorting block on pins 2-3 (WE position.)

4. Install a ZIF socket in the EEPROM socket @ U2. If you have a Rev E1 or earlier PCB, you will have to remove C8A due to it preventing the ZIF socket from seating fully. Replace C8A with a 1uf Tantalum capacitor, which is much smaller. Bend the leads to lay it flat on the PCB, with the long lead in the square pad, then solder. Install the ZIF socket, then re-install the CURRENT EEPROM.

Burning the EEPROM

5. Install J15 on your SmallZ80. This will force the system to remain in Slow speed during initialization. The EEPROM.COM program will abort if the system is not in slow speed.

6. Power up the SmallZ80. Verify that the HiSpd LED @ D5 remains off. Boot the system off the new CF-IDE image. Shift to the A1: User Area and execute the EEPROM.COM program.

7. If the system is still in High speed, EEPROM.COM will abort with an error message.

8. The EEPROM.COM program will ask if it is OK to overwrite the existing OLDEPROM.BIN file. If you press anything but Y, the program will abort. It you want to maintain a copy of the current OLDEPROM.BIN file, you can copy it to something else:

```
COPY OLDEPROM.BIN OLDEPROM.OLD
```

9. The program will prompt you to allow writing the Setup Data Block. Most updates will not require you to write it, but some will. Writing the Setup Data Block will reset all of your custom Setup parameters, including resetting the default terminal to Serial A @ 9600 baud. You should check with TerryG@stack180.com to be certain.

Appendix G
Updating the BootRom (cont)

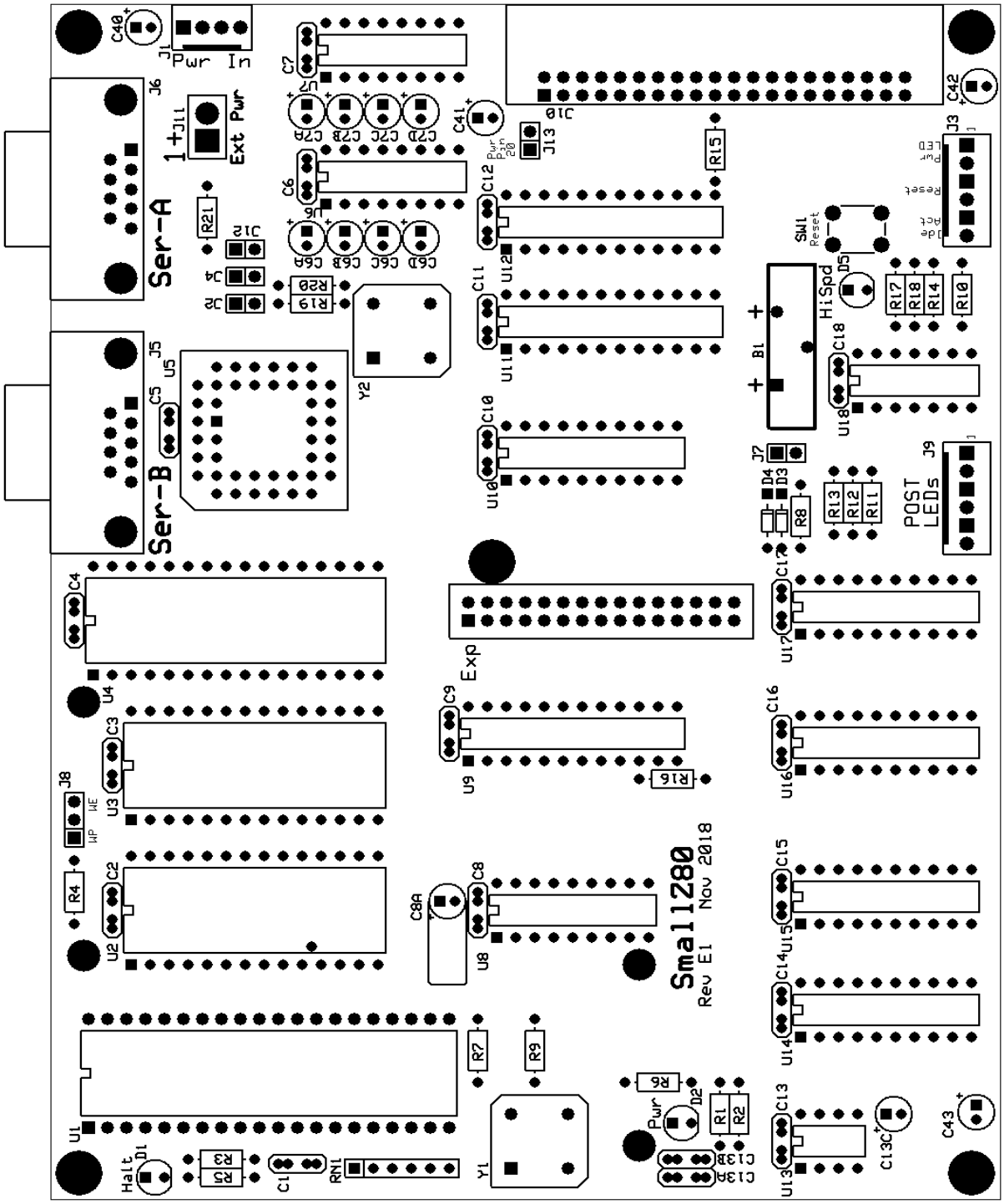
10. When the copy is complete, the program will attempt to read EEPROM.BIN. This is the NEW EEPROM code file. If the file is present and readable, you will be prompted to write the EEPROM. Press Y to begin writing. The Write will take place in four, 8K blocks, just as did the Read.

11. If the system hangs during the 1st block write, you probably neglected to set the Write Enable jumper @ J8 as called for in Step 3 above. You cannot write to the EEPROM when J8 is in the WP (Write Protect) position!

12. When the write is complete and the program exits to the command line, press RESET. Leave the J8 jumper in the WE position. You *should* get a clean re-start. Verify that the "Software Data Prot:" status is TRUE. If you DID write the Setup Data Block, then you may have to reconfigure you terminal program to use Serial A @ 9600 baud since that's what the SmallZ80 will be using after the Setup Data Block write.

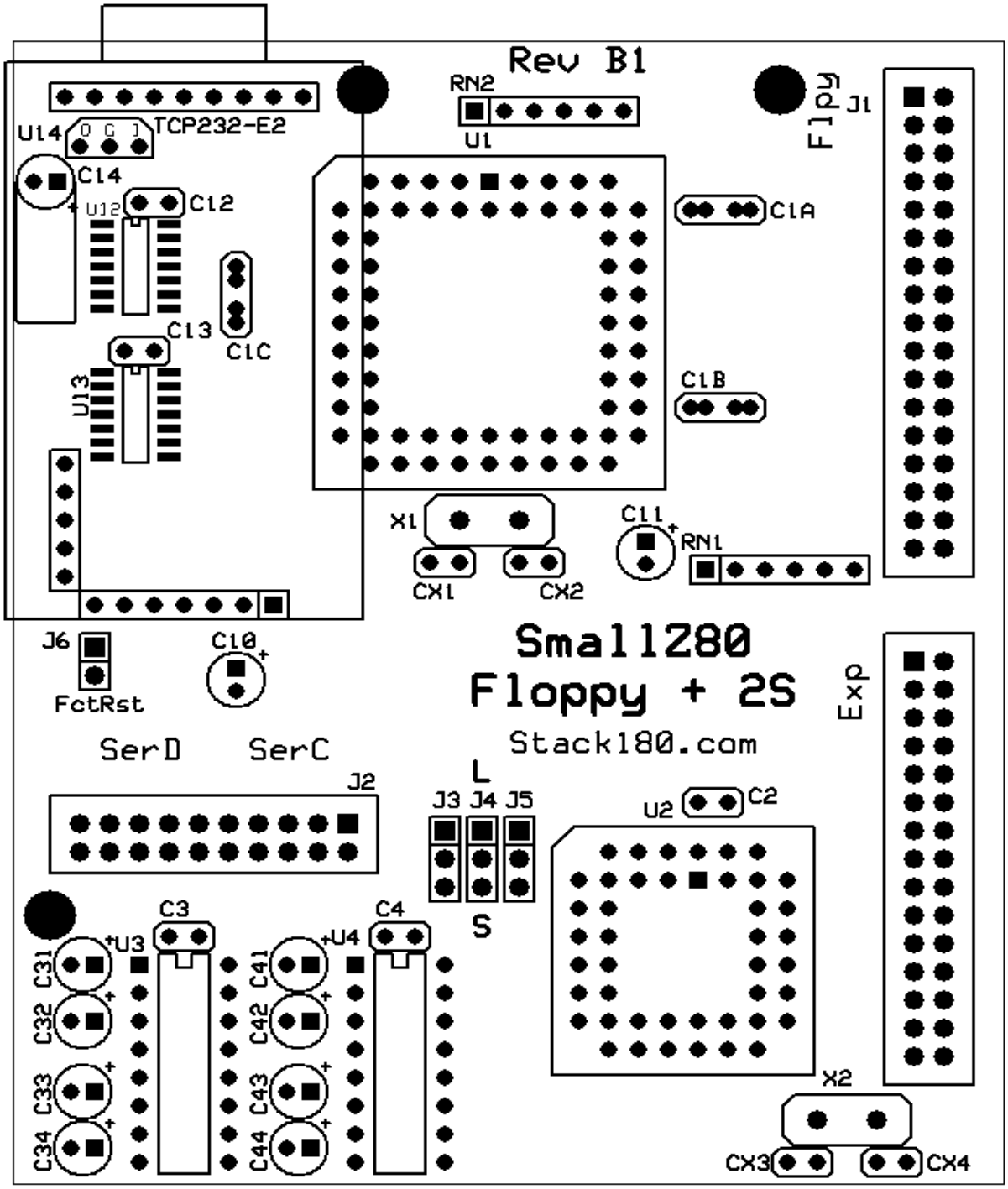
13. Since you just installed a new Distribution EEPROM, you may have to reset all of your custom Setup parameters. If you did not allow the Setup Data Block write, then you should not have to restore your settings. On the other hand, if a Setup Block write was required and you neglected to write it, there WILL be issues with the setup. Most likely, due to the way we add parameters to the Data Block, you will be fine anyway.

Appendix H
Layout and Schematic Diagrams



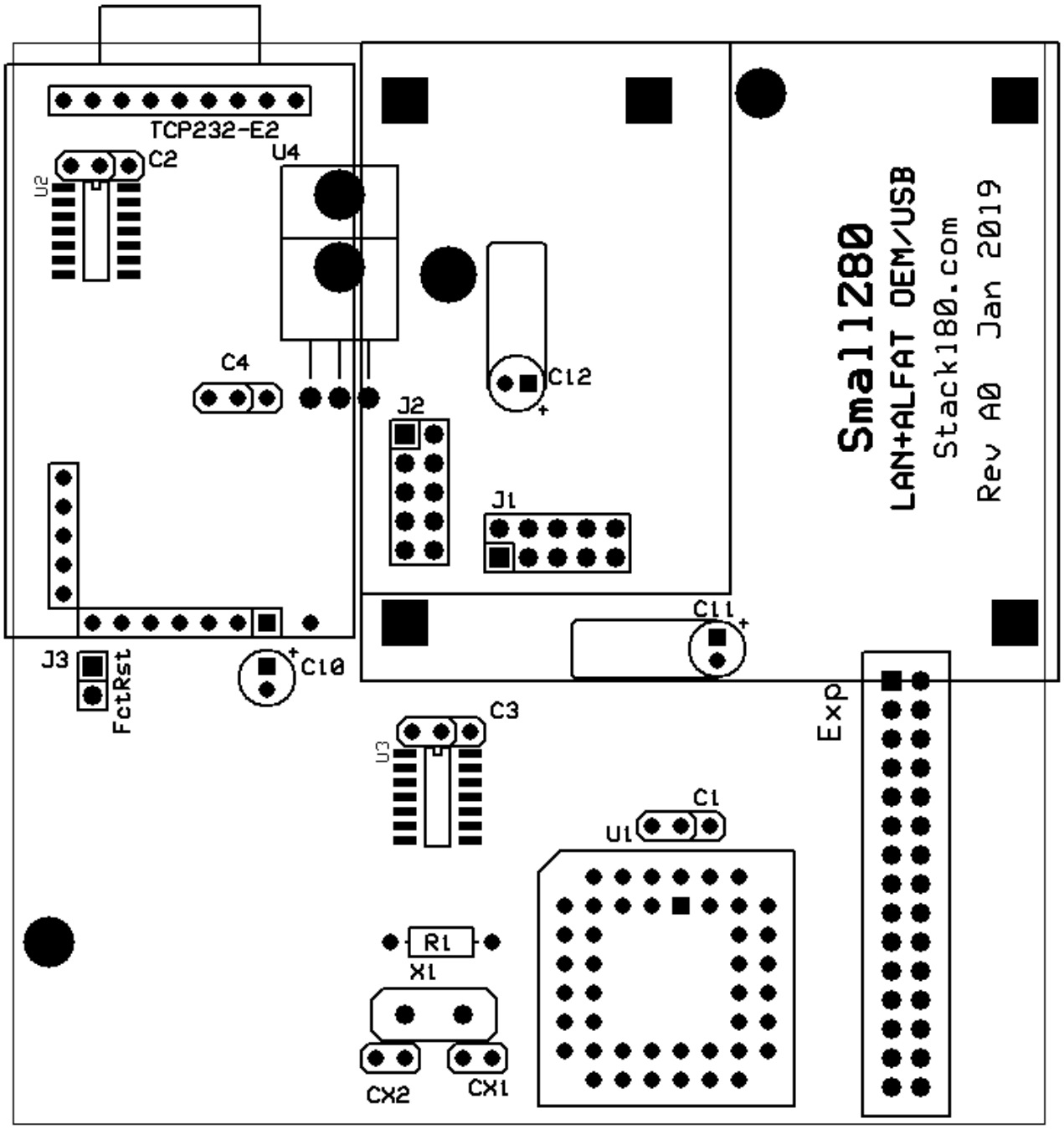
CPU Rev E1 Mechanical Layout (Front)

Appendix H
Layout and Schematic Diagrams (Cont)



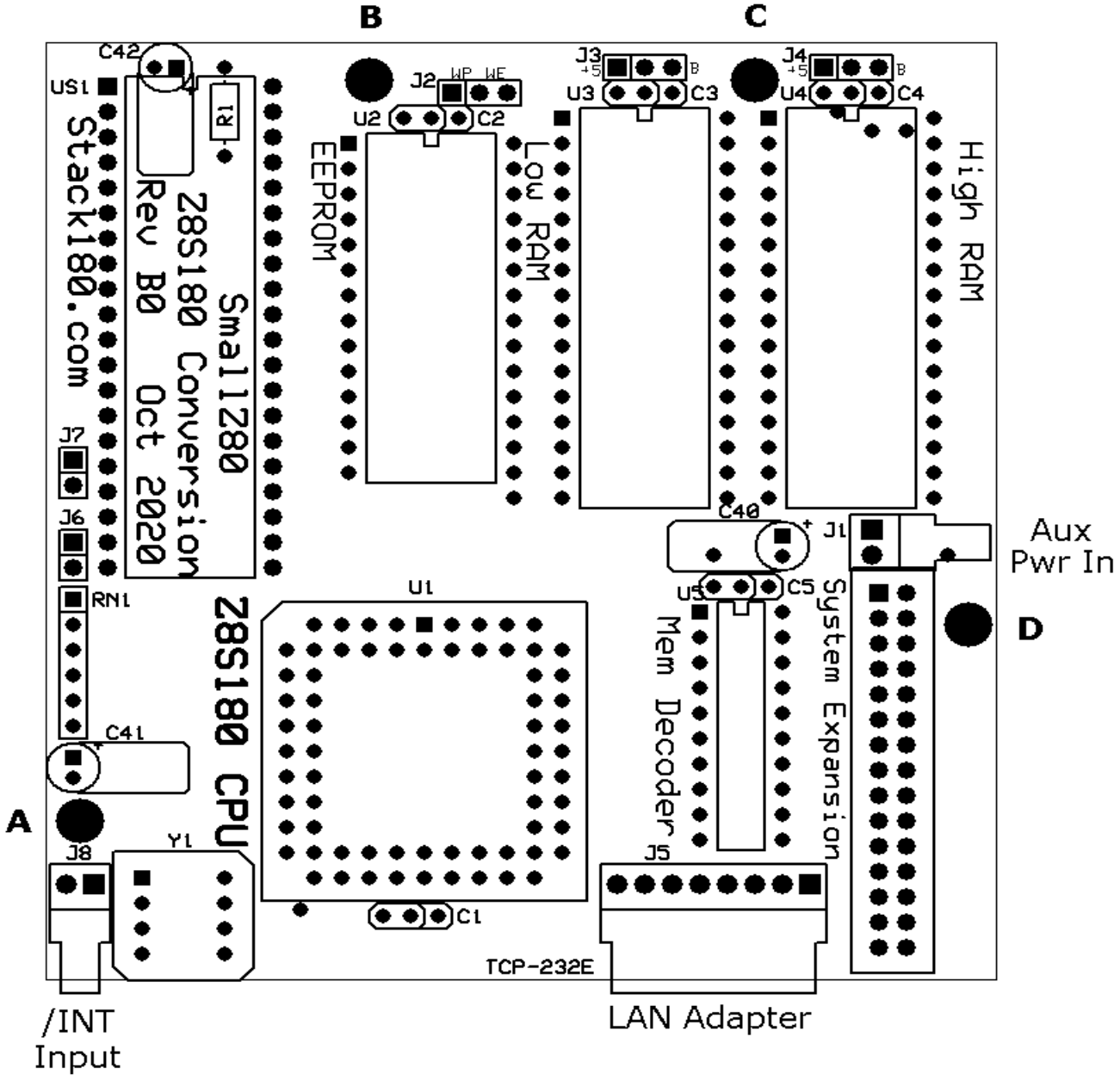
Floppy + 2S Rev B1 Mechanical Layout (Front)

Appendix H
Layout and Schematic Diagrams (Cont)



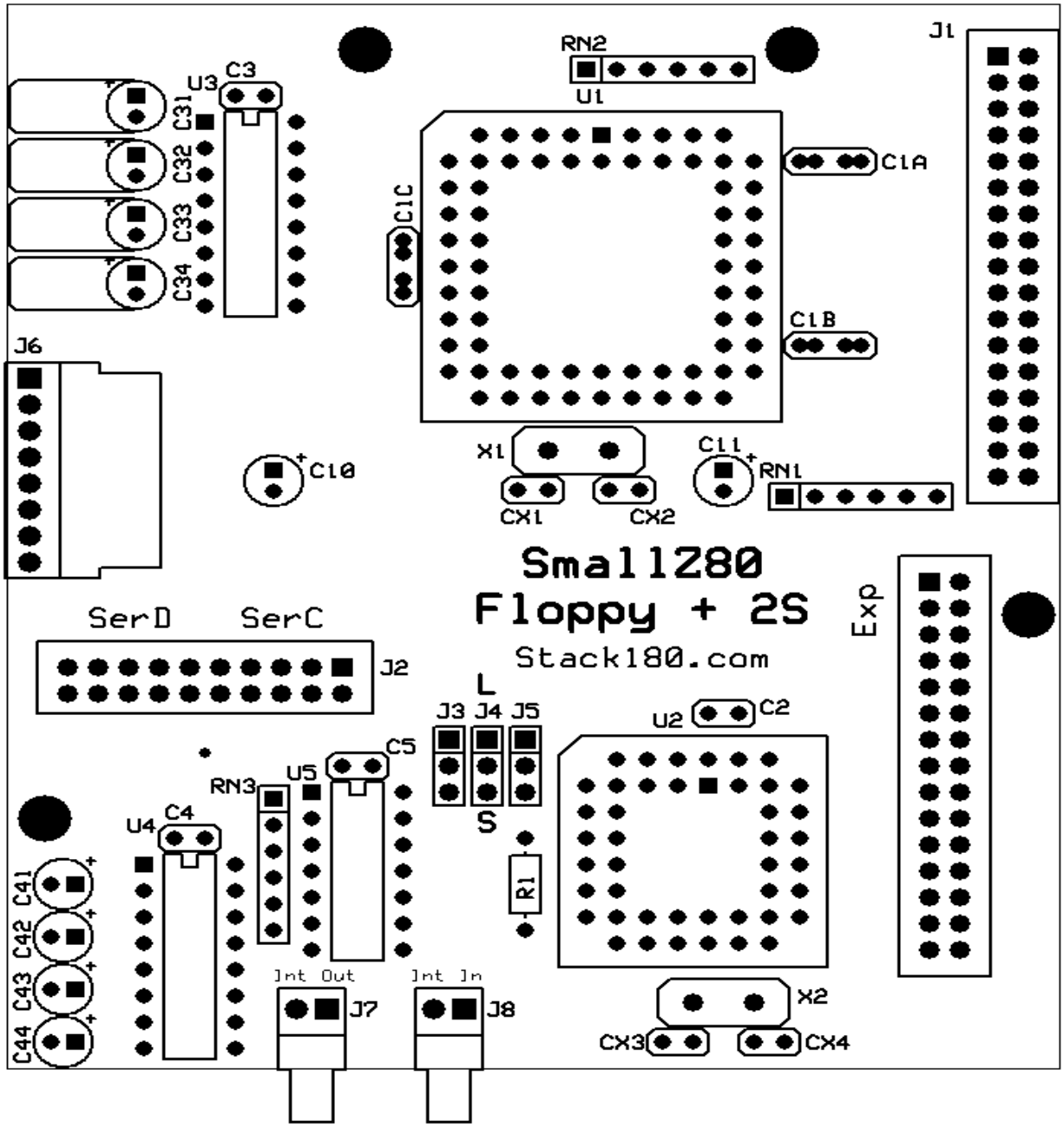
LAN + ALFAT Mechanical Layout (Front)

Appendix H
Layout and Schematic Diagrams (Cont)



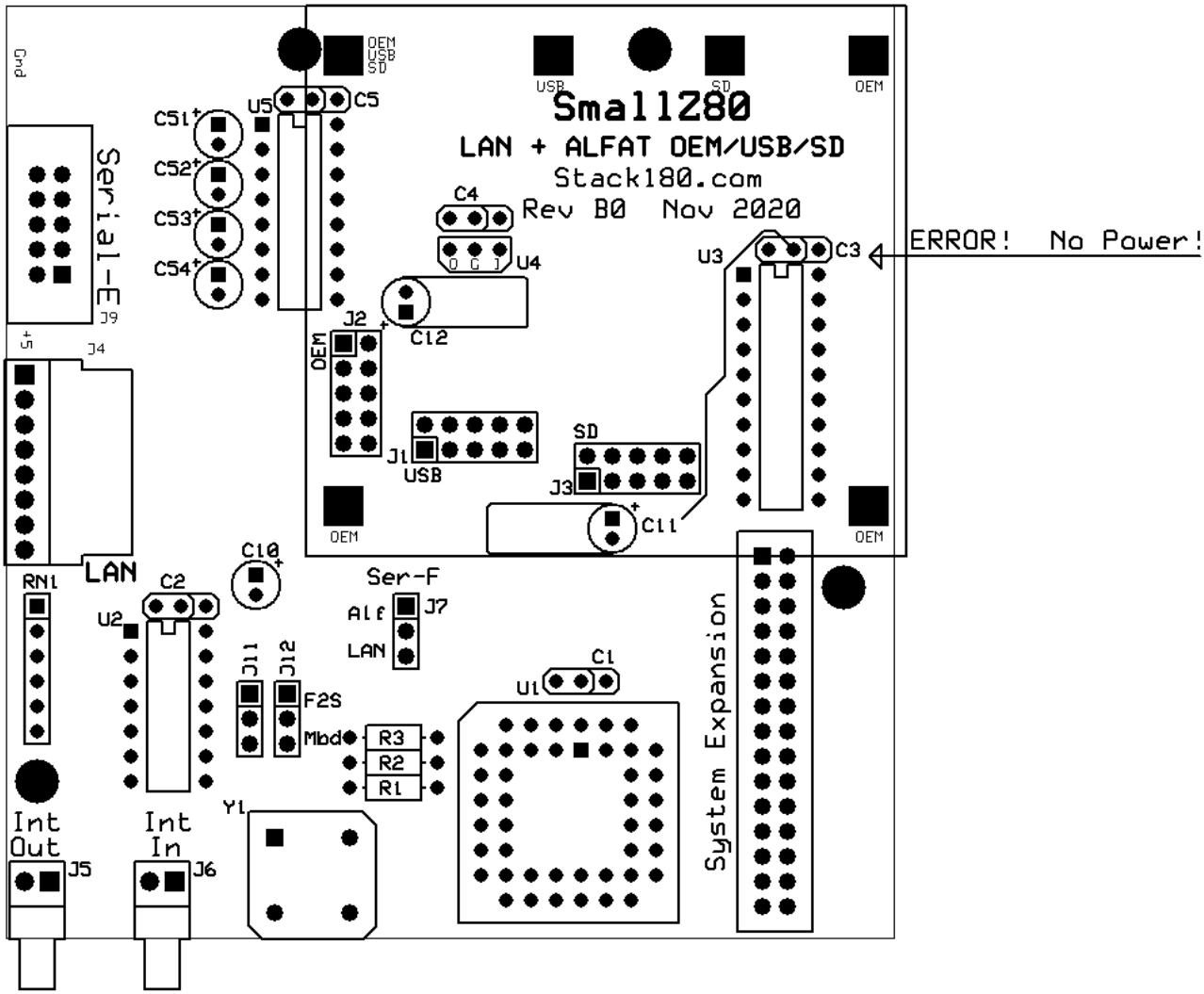
Z180 CPU Adapter Mechanical Layout (Front)

Appendix H
Layout and Schematic Diagrams (Cont)



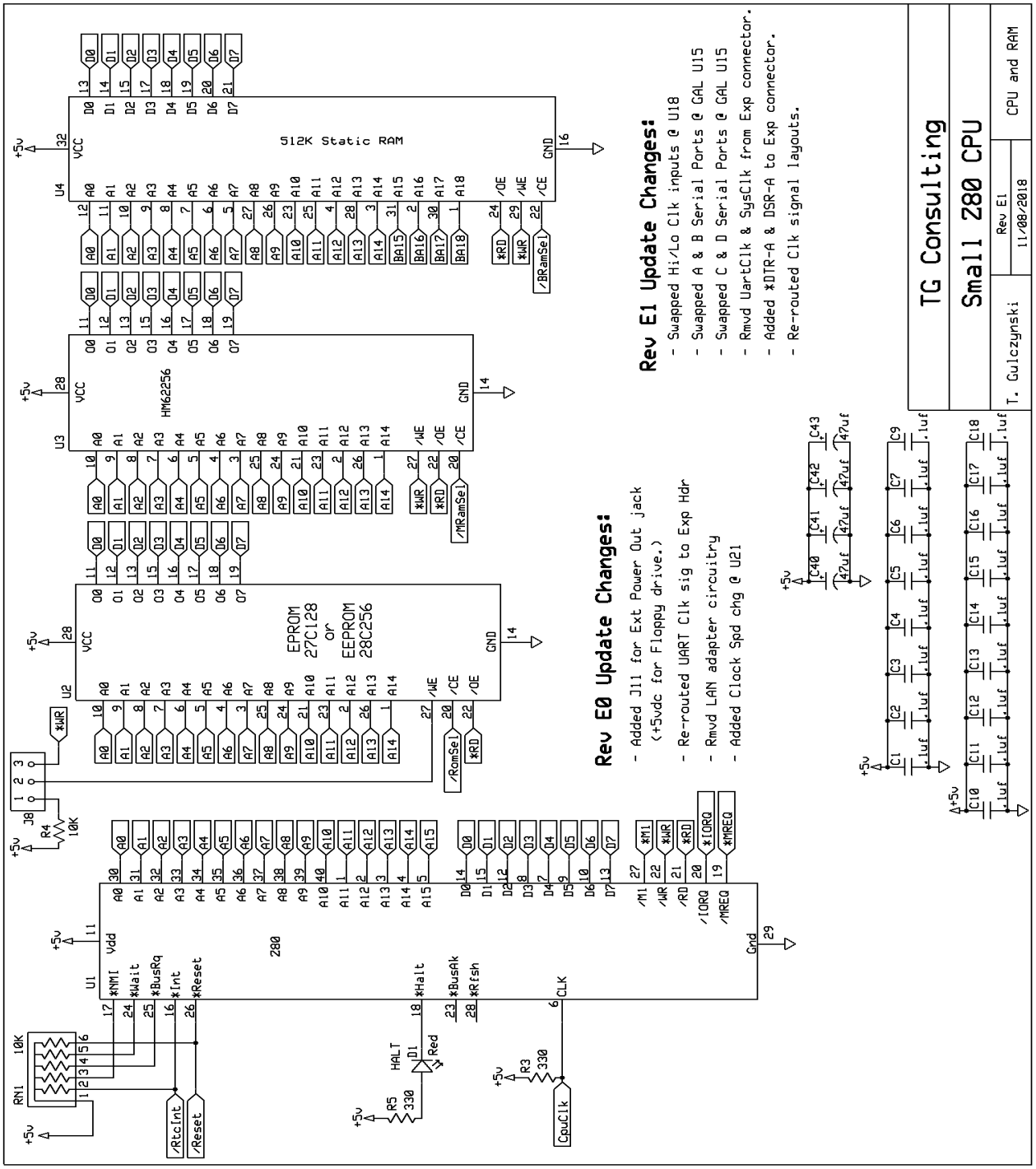
New Style Floppy+2S Adapter Mechanical Layout (Front)

Appendix H
Layout and Schematic Diagrams (Cont)



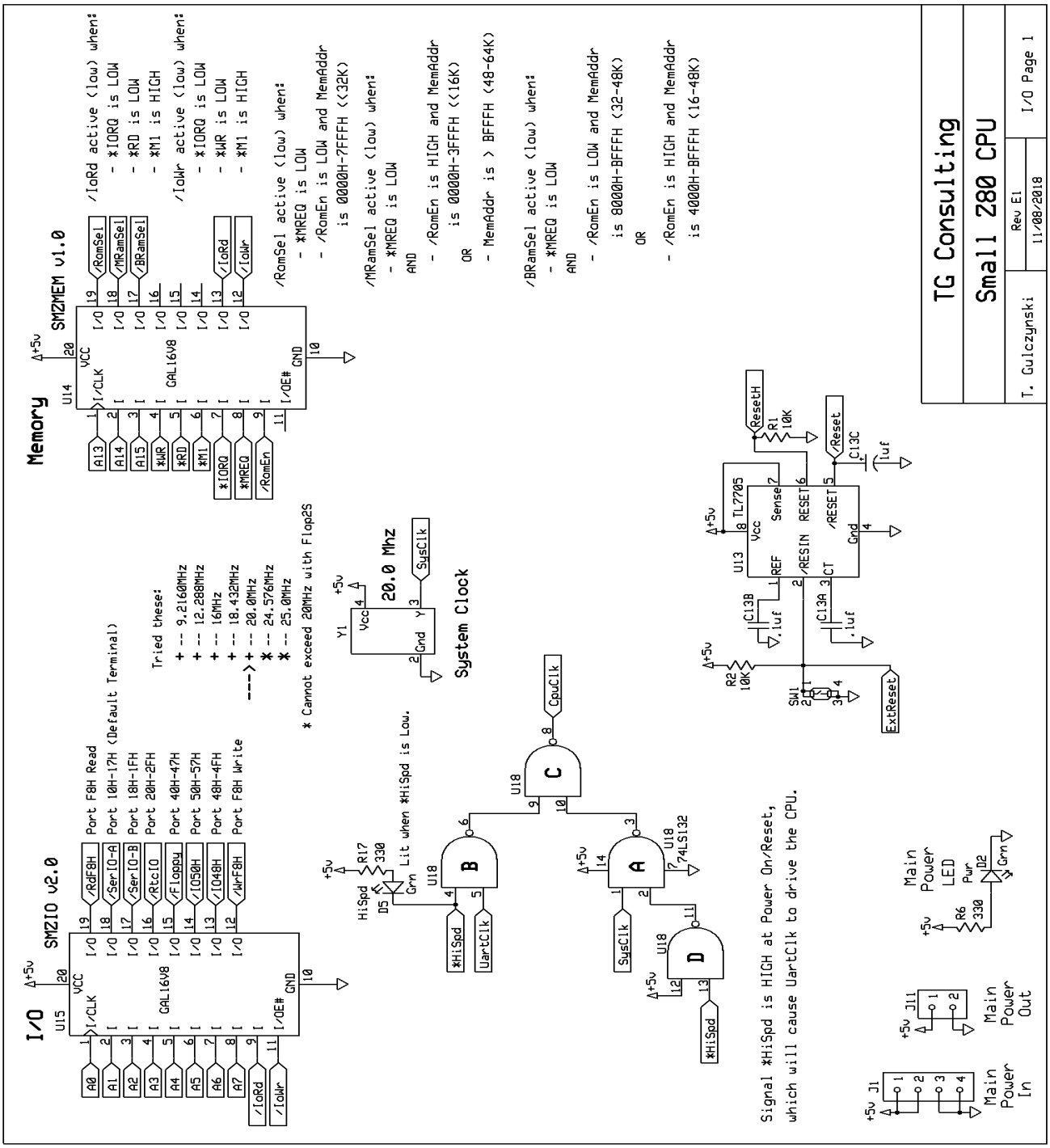
New Style Floppy+2S Adapter Mechanical Layout (Front)

Appendix H
Layout and Schematic Diagrams (Cont)



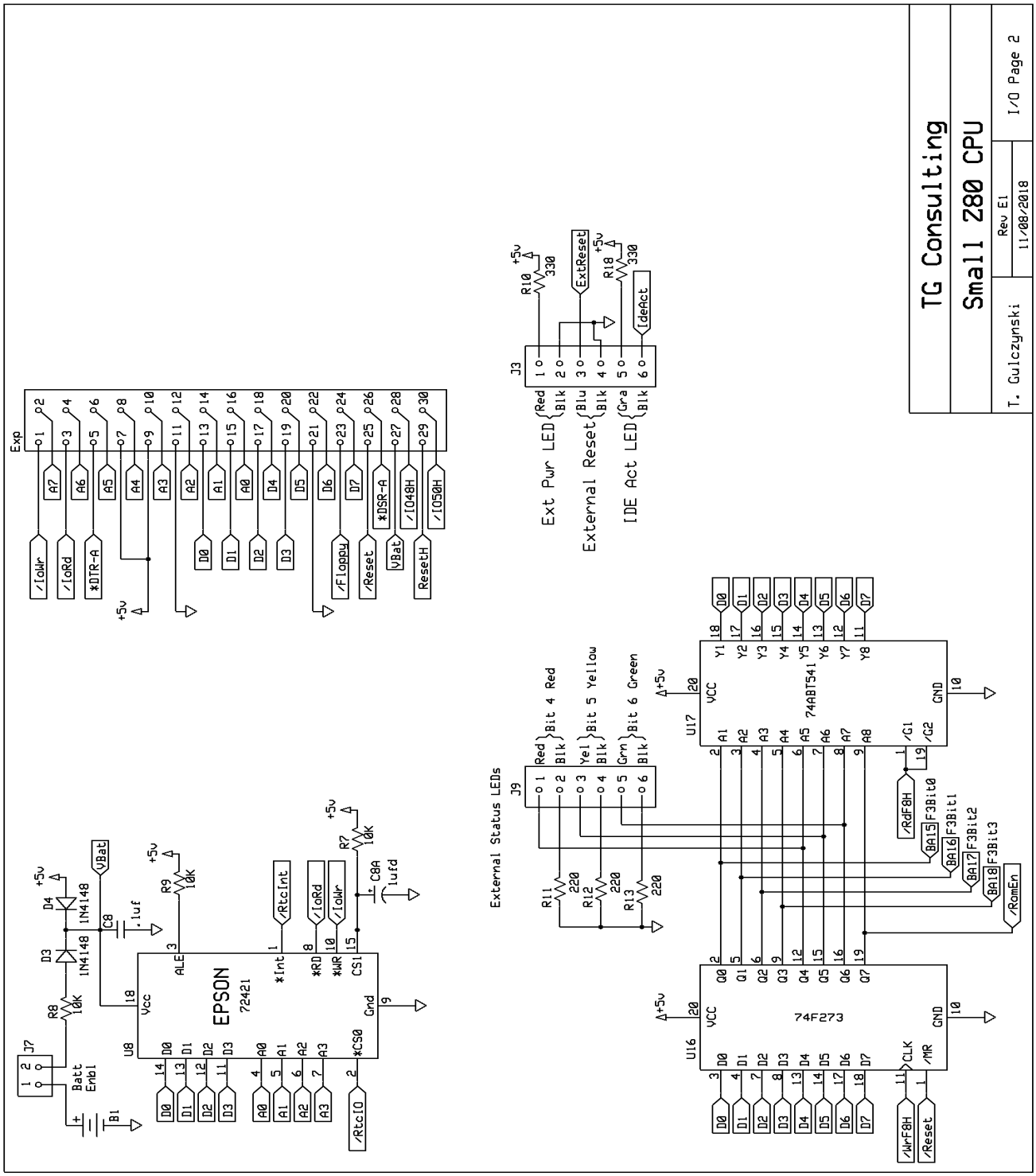
Mainboard Pg 1: CPU and Memory

Appendix H
Layout and Schematic Diagrams (Cont)



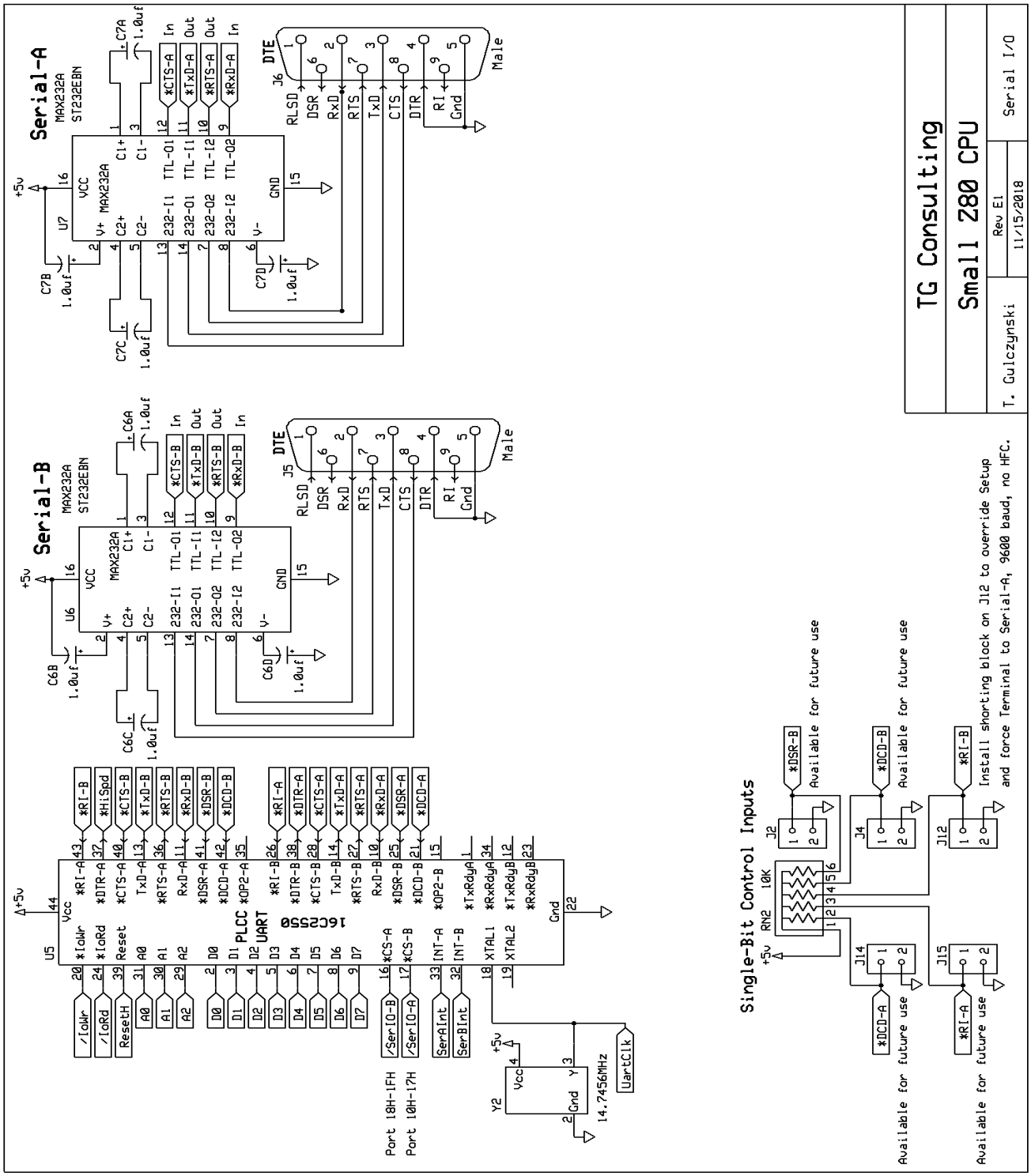
Mainboard Pg2: Memory and I/O Decoder

Appendix H
Layout and Schematic Diagrams (Cont)



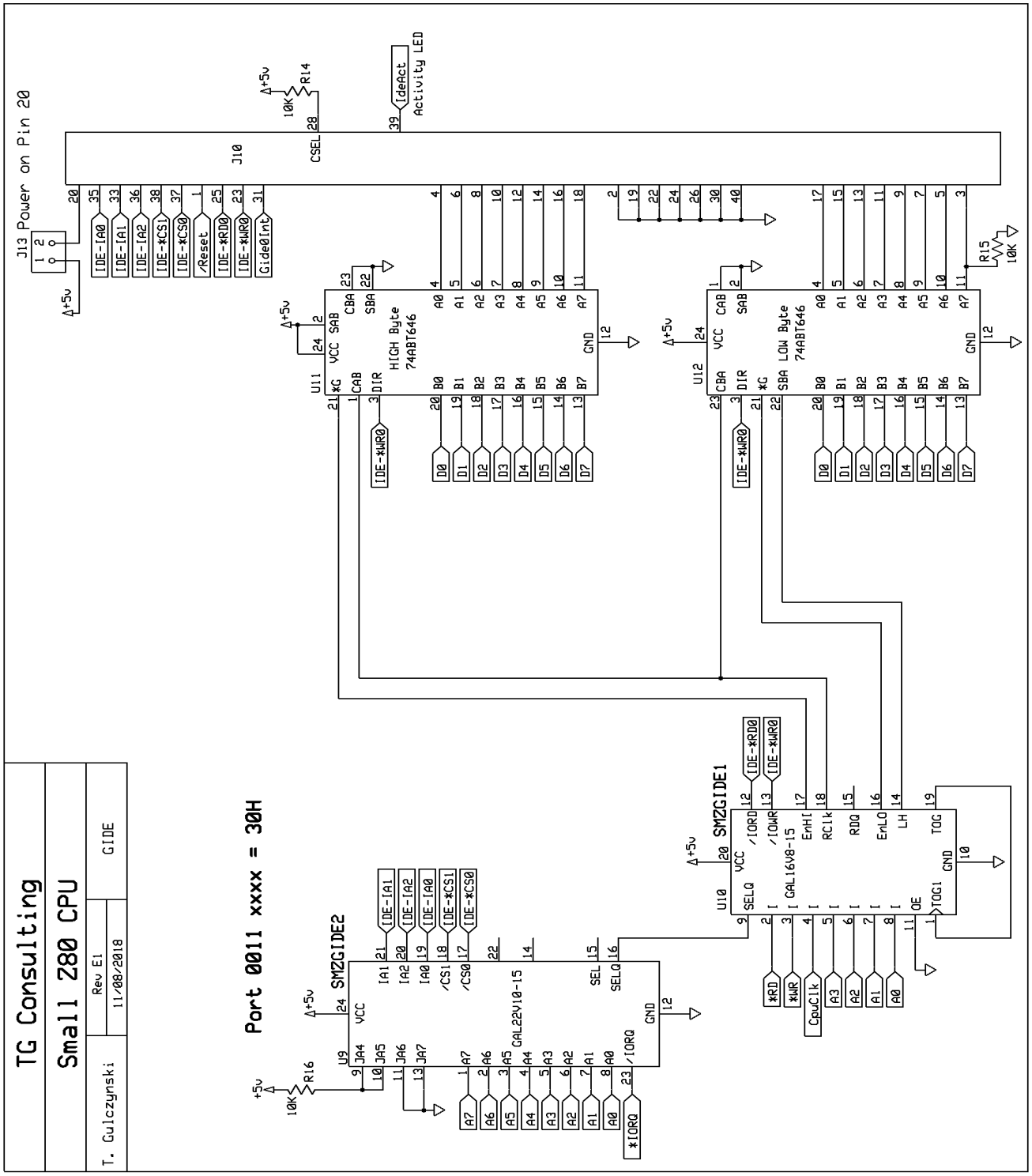
Mainboard Pg3: I/O, RTC & Expansion Bus

Appendix H
Layout and Schematic Diagrams (Cont)



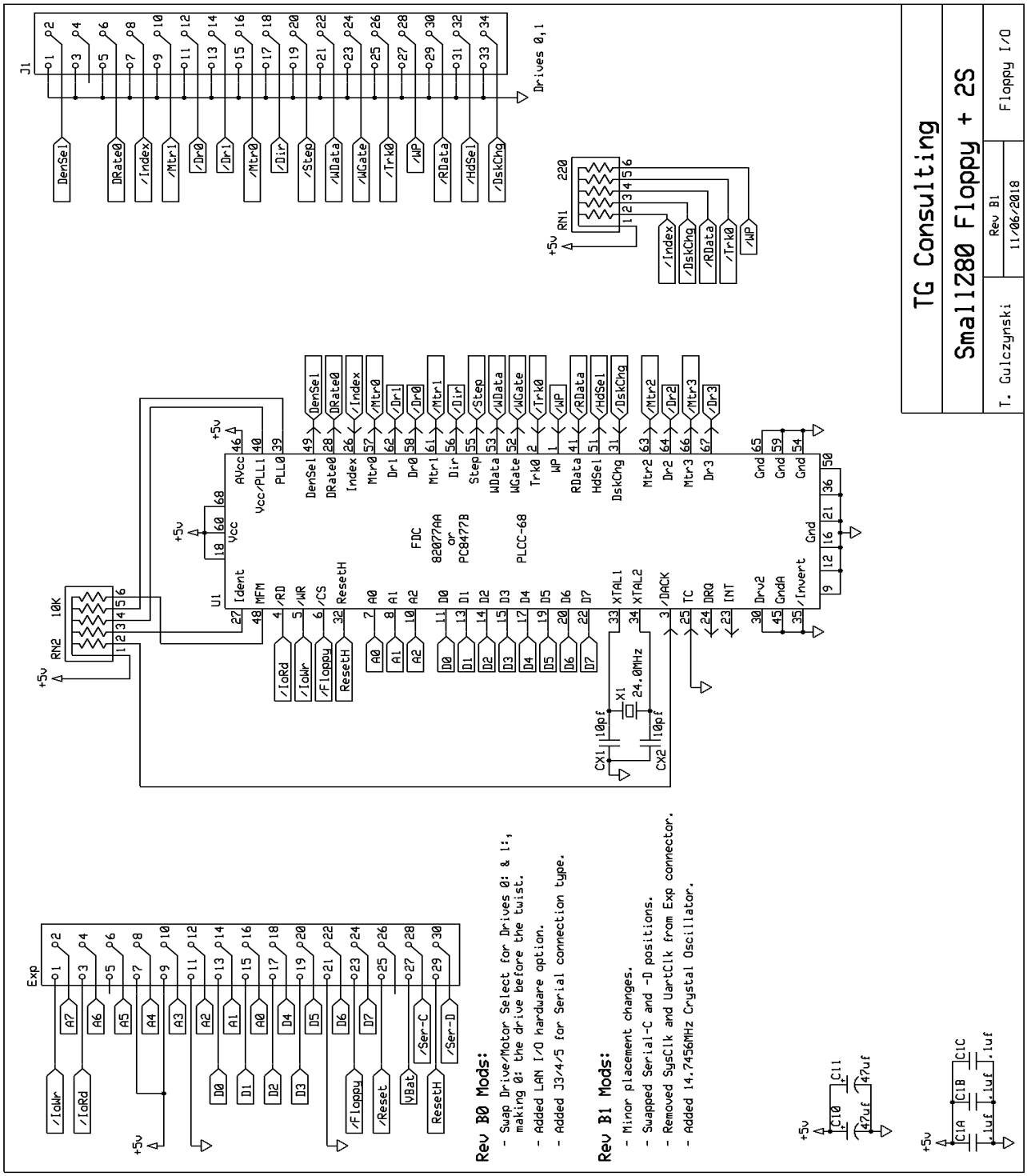
Mainboard Pg 4: Serial I/O

Appendix H
Layout and Schematic Diagrams (Cont)



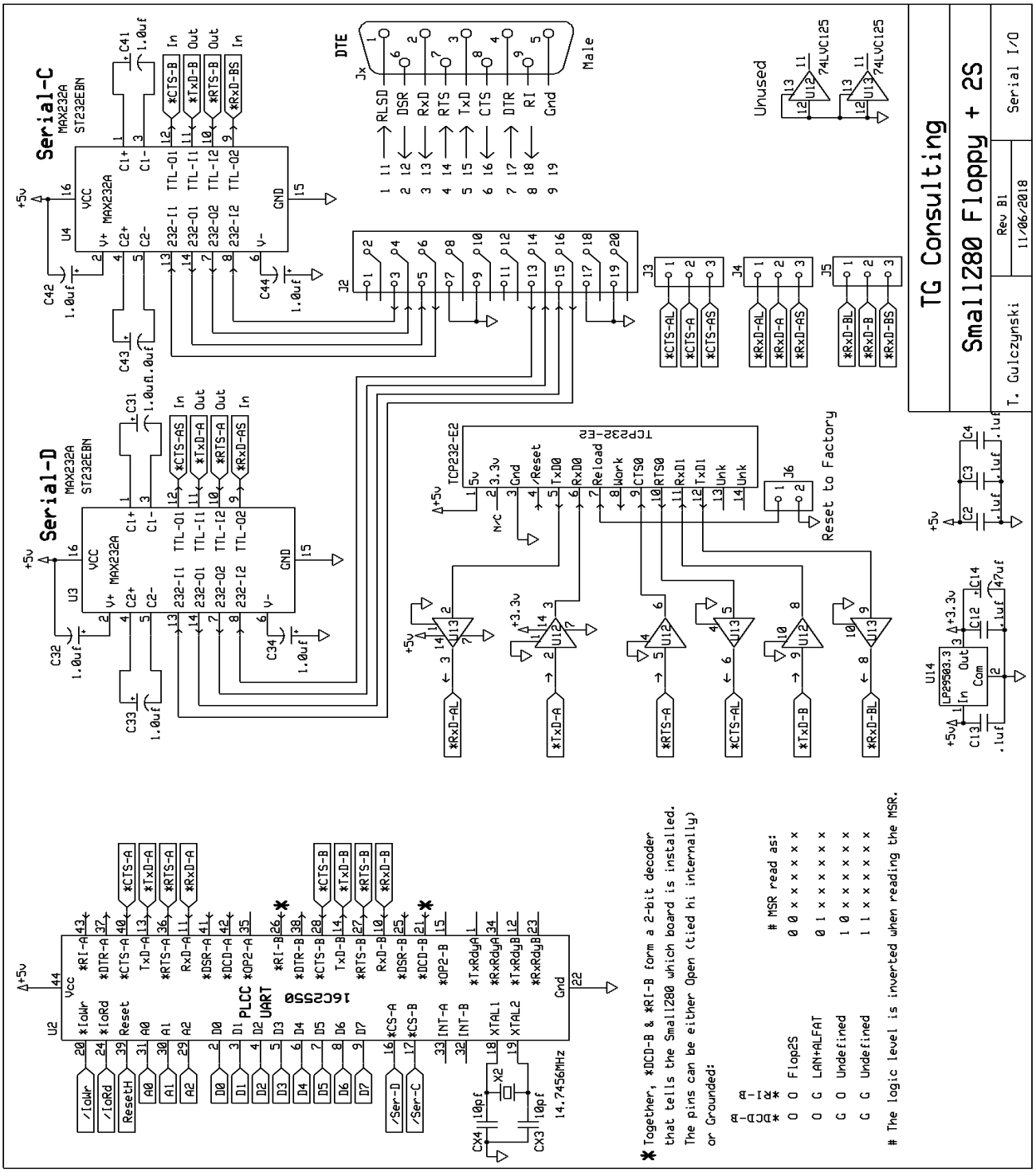
Mainboard Pg 5: Generic IDE

Appendix H
Layout and Schematic Diagrams (Cont)



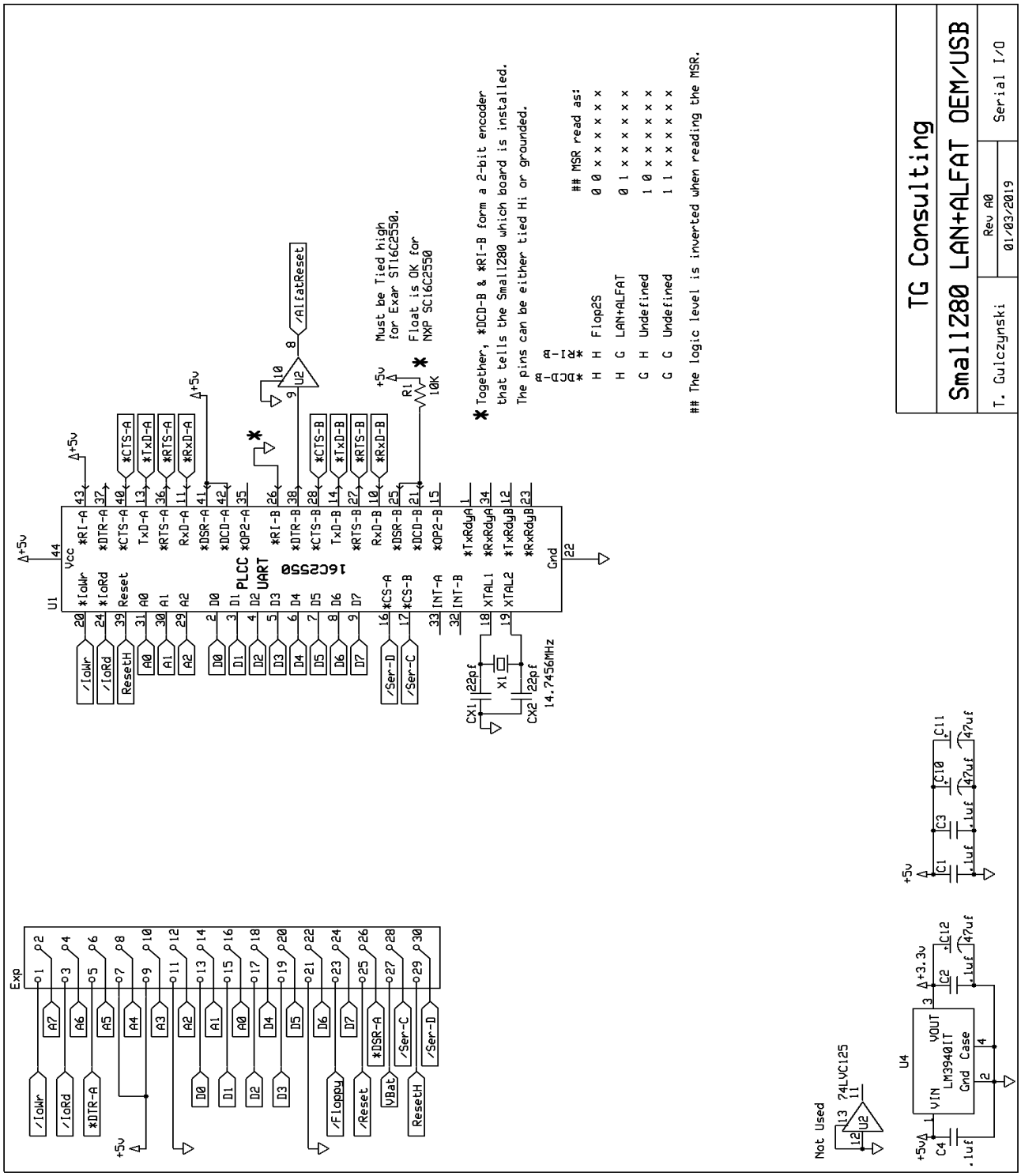
Floppy + 2S Pg 1: Floppy I/O

Appendix H
Layout and Schematic Diagrams (Cont)



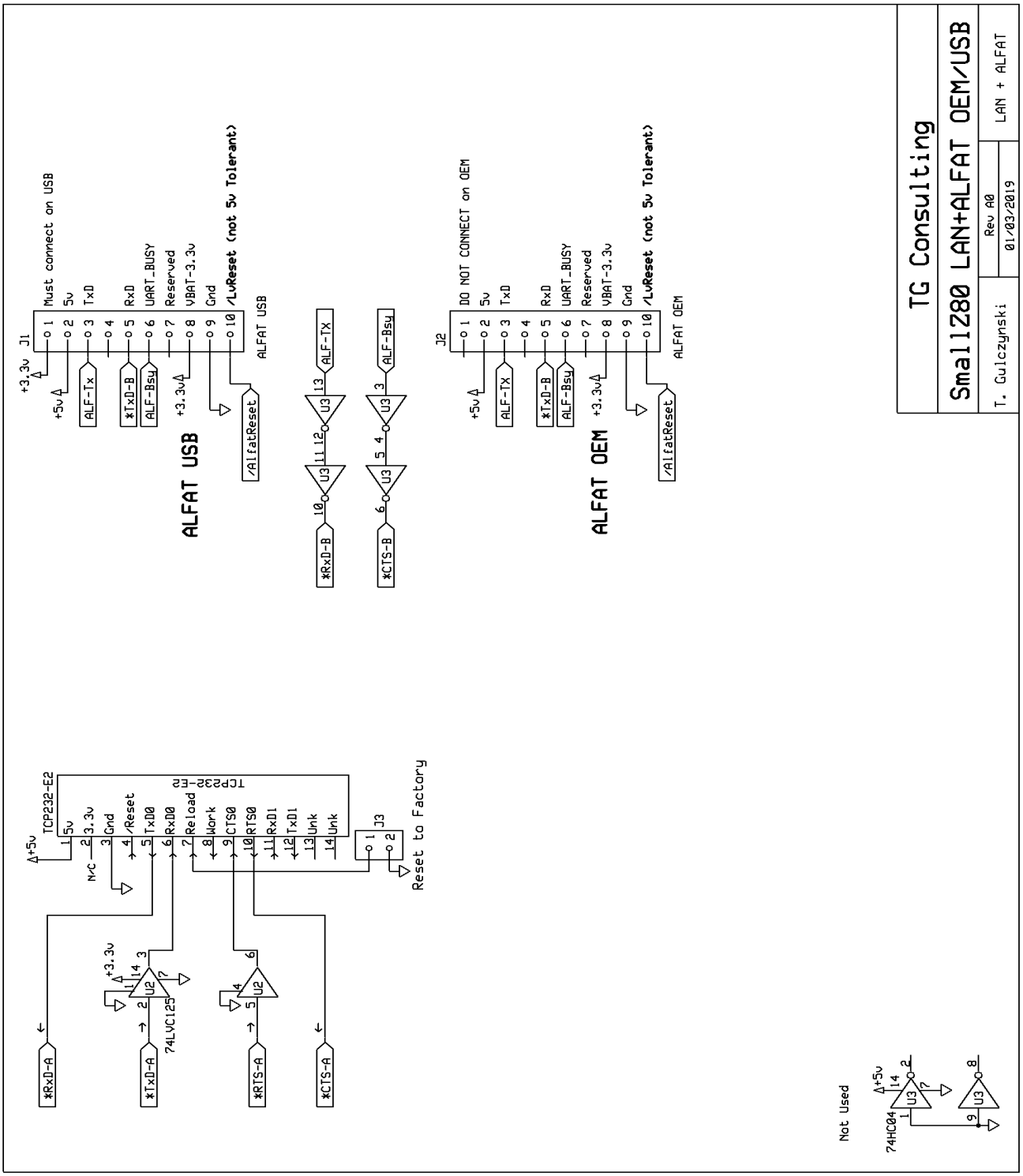
Floppy + 2S Pg 2: Serial I/O

Appendix H
Layout and Schematic Diagrams (Cont)



LAN + ALFAT Pg 1: Serial I/O

Appendix H
Layout and Schematic Diagrams (Cont)



LAN + ALFAT Pg 2: LAN + ALFAT

Appendix I
Parts Lists

SmallZ80 Mainboard

Sockets
1 @ 8-pin socket @ U13
1 @ 14-pin socket at U18
2 @ 16-Pin sockets @ U6 & U7
1 @ 18-Pin socket @ U8
5 @ 20-pin sockets @ U10, U14, U15, U16 & U17
3 @ 24-pin sockets @ U9, U11 & U12
2 @ 28-pin .6” sockets @ U2 & U3
1 @ 32-pin socket @ U4
1 @ 40-pin socket @ U1
1 @ 44-pin PLCC socket @ U5

Integrated Circuits
1 @ Z80 CPU – 20MHz @ U1
1 @ 28C256-150 EEPROM @ U2 - pre-programmed
1 @ AS6C62256-55 32K x 8 Static RAM @ U3
1 @ AS6C4008-55 (628512-55) 512K x 8 Static RAM @ U4
1 @ 16C2550 PLCC-44 UART @ U5
2 @ MAX232 RS-232 level shifters @ U6 & U7
1 @ Epson 72421B RTC @ U8
1 @ 22V10-15 GAL @ U9 – Labeled SMZ-GIDE-2/60 – pre-programmed
1 @ 16V8-15 GAL @ U10 – Labeled SMZ-GIDE-1 – pre-programmed
2 @ 74F646 (or 74HC646 or 74ALS646 or 74ABT646) @ U11 & U12
1 @ TL7705 Reset Generator @ U13
1 @ GAL16V8-10 – labeled SMZ-Mem v1.0 @ U14 – pre-programmed
1 @ GAL16V8-15 – labeled SMZ-IO v3.0 @ U15 – pre-programmed
1 @ 74F273 (or ALS273) @ U16
1 @ 74ABT541 (or F541 or ALS541) @ U17
1 @ 74HC132 at U18 (HC is preferred, LS will work)

Resistors
2 @ 6-pin, 10K resistor array (labeled '103') @ RN1, RN2
6 @ 330 Ohm ¼ watt resistors (Org, Org, Brn) @ R3, R5, R6, R10, R17, R18
9 @ 10K Ohm ¼ watt resistors (Brn, Blk, Org)R1, R2, R4, R7/8/9, R14/15/16
3 @ 220 Ohm ¼ watt resistors (Red, Red, Brn) @ R11, R12, & R13

Capacitors
20 @ .1ufd Mono bypass caps @ C1 – C18, C101 & C102
4 @ 47ufd electrolytic capacitors @ C40-C43
10 @ 1.0uf Electrolytic caps @ C8A, C6A-C6D, C7A-C7D & C13C

Appendix I
Parts Lists (cont.)

SmallZ80 CPU (cont.)

Diodes
2 @ Red T1¾ LED @ HALT & External POST
4 @ Green T1¾ LED @ Power, HiSpd, External POST & External Pwr
2 @ 1N4148 diodes @ D3 & D4
2 @ Yellow T1¾ LED @ External POST & External IDE Activity

Connectors
2 @ 6-pin keyed headers @ J3 & J9
7 @ 2-pin non-keyed header @ J2, J4, J7, J12, J13, J14 & J15
1 @ 3-pin non-keyed header @ J8
1 @ 4-pin keyed header @ J1 (main power input)
1 @ 2x15 (30-pin) female expansion header @ Exp
1 @ 40-pin right angle IDE header @ J10
2 @ 9-pin Male RS-232 connectors @ J5 & J6
1 @ Power Input 4-pin female connector (w/8” leads)
1 @ POST LED 6-pin socket header w/ 8” leads
1 @ Pwr+IDE+Reset 6-pin socket header w/ 8” leads

Oscillators
1 @ 20.0MHz @ Y1
1 @ 14.7456MHz @ Y2

Miscellaneous
10 @ 1/4” 4-40 machine screws
5 @ 1/2” 4-40 F-F hex standoffs
6 @ shorting blocks @ J2 / J4 / J7 / J8 / J13 + one extra
1 @ Battery socket
1 @ CR2032 Lithium battery
1 @ OMRON B3F-1000 SPST push button Reset switch @ SW1
1 @ Dual CF-IDE adapter
1 @ 64MB CF module, loaded (may be 128MB)
1 @ PCB
1 @ CDROM with instructions, schematics, diagrams, software, etc

SmallZ80 Clock Control Module Addendum

1 @ Clock Control PCB
2 @ 14-pin sockets @ U18, U19
1 @ 74HC132 @ U18
1 @ 74LS92 @ U19
1 @ Green LED @ D1
1 @ 330 Ohm resistor (Org, Org, Brn) @ R1
2 @ .1uf mono bypass cap
1 @ 2x3 pin header
1 @ shorting block
2 @ 1x7 extension pin headers

Appendix I
Parts Lists (Cont.)

Floppy + 2S (Old Style)

Sockets
1 @ 68-pin PLCC at U1
1 @ 44-pin PLCC at U2
2 @ 16-pin at U3 & U4

Integrated Circuits
1 @ 82077AA or PC8477B (PLCC-68) at U1
1 @ NXP 16C2550 Dual UART (PLCC-44) at U2 (Exar requires PCB mod.)
2 @ MAX232 at U3 & U4
2 @ 74LVC125 at U12 & U13 (SOIC SMD - pre-installed)

Capacitors
8 @ 1uf Electrolytic at C31-C34 and C41-C44
4 @ 10pf monolithic caps at CX1, CX2, CX3 & CX4
6 @ .1uf monolithic bypass caps at C1A-C1C & C2-C4
2 @ 47ufd Electrolytic at C10, C11

Resistors
1 @ 220 x 6pin network (labeled '221') at RN1
1 @ 10K x 6pin network (labeled '103') at RN2

Connectors & Headers
1 @ 2x15 long pin header at Exp
1 @ 2x17 straight box header at J1 (Floppy port)
1 @ 2x10 straight box header at J2 (serial I/O connector)
3 @ 1x3 pin headers at J3, J4, & J5

Miscellaneous
3 @ 1/2” 4-40 F-F hex standoffs
6 @ 1/4” 4-40 machine screws
1 @ 24MHz crystal at X1
1 @ 14.7456MHz crystal at X2
1 @ Dual 9-pin male DTE pigtail (6” ribbon with 2x10 female header)
3 @ Shorting blocks
1 @ Floppy + 2S Adapter PCB

CDROM in SmallZ80 CPU kit, with instructions, schematics, diagrams, software, etc.

TCP232 LAN Adapter Option
1 @ 1x7 Socket Header at TCP232
1 @ 1x5 socket header at TCP232
1 @ 1x9 socket header at TCP232
2 @ .1uf monolithic bypass caps at C12, C13
1 @ 47ufd Electrolytic at C14
1 @ 1x2 pin header at J6 (Factory Reset)
1 @ LP29503.3 Regulator @ U14
1 @ TCP232-E (or -E2) Ethernet LAN Adapter

Appendix I
Parts Lists (Cont.)

LAN + ALFAT (OEM / USB / SD) (Old Style)

Common to all configurations

Sockets
1 @ 44-pin PLCC at U1

Integrated Circuits
1 @ NXP 16C2550 Dual UART (PLCC-44) at U1
1 @ 74LVC125 at U2 (SOIC SMD – pre-installed)
1 @ 74HCT04 at U3 (SOIC SMD - pre-installed

Capacitors
2 @ 22pf monolithic caps at CX1 & CX2
2 @ .1uf monolithic bypass caps at C1 & C4
3 @ 47ufd Electrolytic at C10, C11, C12

Resistors
1 @ 10K resistor (Brn, Blk, Org) at R1

Connectors & Headers
1 @ 2x15 short pin header at Exp

Miscellaneous
1 @ 14.7456MHz crystal at Y1
1 @ LAN + ALFAT PCB
1 @ LM3940IT 3.3v Regulator (TO-92)
1 @ 3/8” x 4-40 machine screw
1 @ 4-40 machine nut
2 @ 12mm F-F hex standoffs
4 @ 3mm x 6mm machine screws

TCP232 LAN Option

1 @ Shorting block
1 @ 1x2 pin header at J3
1 @ .2uf Mono bypass cap @ C2
1 @ 5-, 7-, & 9-pin socket headers
1 @ tie wrap
1 @ TCP232 LAN daughterboard adapter (with 5/7/9 pin headers installed)

ALFAT OEM / USB / SD Card Adapter Option

1 @ .1uf mono bypass cap @ C3
1 @ 2x5 socket header @ J1 (OEM) or J2 (USB) or J3 (SD Card)
1 @ ALFAT daughterboard adapter (OEM or USB or SD Card)
Standoff hardware (Standoff + 2 screws) (2 sets for USB/SD, 4 sets for OEM)

Appendix I
Parts Lists (Cont.)

Z8S180 CPU Adapter (Z180 Upgrade)

Sockets
1 @ 68-pin PLCC @ U1
1 @ 28-pin DIP socket @ U2
2 @ 32-pin DIP socket @ U3, U4
1 @ 20-pin DIP socket @ U5

Integrated Circuits
1 @ Z8S180-33 CPU (PLCC-68) at U1
**1 @ Zicor X28C256 or Atmel 28C256 EEPROM @ U2 (Optional)
**1 & AS6C4008-55 512Kx8 Static RAM @ U3 & U4 (2nd RAM is optional)
**1 @ GAL16V8-5 @ U5 (pre-programmed, labeled SMZ180-Mem) (optional)

Capacitors
5 @ .1uf monolithic bypass caps at C1 - C5
3 @ 100ufd Electrolytic caps @ C40, C41 & C42
1 @ 10uf Tantalum cap @ C43

Resistors
1 @ 10K resistor (Brn, Blk, Org) at R1
1 @ 330 resistor (Org, Org, Brn) at R2 (Rev C0 and later)
1 @ 10K, 6-pin resistor network @ RN1

Connectors & Headers
1 @ 2x15 extended pin mezzanine header at Exp
3 @ 3-pin male header @ J2, J3 & J4
1 @ 2-pin male header @ J6
2 @ 2-Pin keyed, right angle pin header @ J1 & J8
1 @ 8-pin keyed, right angle pin header @ J5
2 @ 1x20-pin short pin headers
1 @ 2-pin Z180 extension power cable 22ga (8”)
1 @ 2-pin (.156”) header and housing for mainboard power (6” 18ga wire)

Miscellaneous
1 @ 22.1184MHz oscillator at Y1 (on mainboard)
1 @ T1¾ Grn LED @ D1 (Rev C0 and later)
3 @ Shorting blocks
1 @ Z180 CPU PCB
1 @ Z80 Riser kit (40-pin socket, 2 @ 1x20 pin headers, PCB)
1 @ CDROM

Hardware
All told, this hardware is used for BOTH Mainboard AND Z180 CPU board:
8 @ 12mm F-F standoffs
1 @ 12mm M-F standoff
16 @ 3-0.5x6mm machine screws

Appendix I
Parts Lists (Cont.)

Floppy + 2S (New Style, Rev B3)

Sockets
1 @ 68-pin PLCC at U1
1 @ 44-pin PLCC at U2
2 @ 16-pin at U3 & U4
1 @ 14-pin at U5

Integrated Circuits
1 @ 82077AA or PC8477B (PLCC-68) at U1
1 @ NXP 16C2550 Dual UART (PLCC-44) at U2 (Exar should be OK)
2 @ MAX232 at U3 & U4
1 @ 74ALS05 at U5 (any version 74xx05 will probably suffice)

Capacitors
8 @ 1uf Tantalum at C31-C34 and C41-C44
4 @ 22pf monolithic caps at CX1, CX2, CX3 & CX4
7 @ .1uf monolithic bypass caps at C1A-C1C & C2-C5 (4 must be 2.54 LS)
2 @ 47ufd Electrolytic at C10 & C11

Resistors
1 @ 10K at R1
2 @ 220 x 6-pin network (labeled '221') at RN1 & RN3
1 @ 10K x 6-pin network (labeled '103') at RN2

Connectors & Headers
1 @ 2x15 short expansion pin header at Exp (+1 for high mount)
1 @ 2x17 straight box header at J1 (Floppy port)
1 @ 2x10 straight box header at J2 (Dual Serial I/O port)
3 @ 1x3 pin headers at J3, J4, & J5
2 @ 2-pin keyed headers (R/A) @ J7 & J8
1 @ 8-pin keyed RA header for output to TCP-232

Miscellaneous
4 @ 12mm or 25mm M-F Standoffs (12mm is typical) (25mm for high mount)
1 @ 24MHz crystal at X1
1 @ 14.7456MHz crystal at X2
1 @ Dual 9-pin male DTE pigtail (8” ribbon with 2x10 female header)
3 @ Shorting blocks
1 @ Floppy + 2S Adapter (New Style) PCB
1 @ Input Interrupt wire harness (12” orange & brown cable, 26ga)
1 @ Output Interrupt wire harness (4” orange & brown cable, 26ga)

Appendix I
Parts Lists (Cont.)

LAN + ALFAT (OEM / USB / SD) (New Style)

Common to all configurations

Sockets
1 @ 44-pin PLCC at U1
1 @ 14-pin DIP socket @ U2
1 @ 20-pin DIP socket at U3
1 @ 16-pin DIP socket @ U5

Integrated Circuits
1 @ NXP 16C2550 Dual UART (PLCC-44) at U1
1 @ 74ALS05 at U2 (most any version of 74xx05 is fine)
1 @ GAL16V8-25 at U3 (pre-programmed)
1 @ MAX232 RS-232 Level Converter at U5

Capacitors
4 @ .1uf monolithic bypass caps at C1, C2, C3 & C5
2 @ 47ufd Electrolytic at C10, C11

Resistors
1 @ 10K resistor (Brn, Blk, Org) at R1
1 @ 2.2K resistor (Red, Red, Red) at R2
1 @ 3.3K resistor (Org, Org, Red) at R3
1 @ 220 6-pin resistor network (labeled '221')

Connectors & Headers
1 @ 2x15 long pin & socket header at Exp
1 @ 10-pin box header at J9 (Serial-E I/O)
2 @ 2-pin RA keyed pin header at J5 & J6 for interrupts
3 @ shorting blocks

Miscellaneous
1 @ 14.7456MHz Oscillator at Y1
1 @ LAN + ALFAT PCB
4 @ 12mm M-F hex standoffs

TCP232 LAN Option

1 @ 8-pin RA header at J4
1 @ TCP-232 LAN Remote adapter
1 @ 12” LAN Extension cable

ALFAT OEM / USB / SD Card Adapter Option

1 @ LP2950 3.3v Regulator (TO-92)
1 @ .1uf mono bypass cap @ C4
1 @ 47uf Electrolytic cap at C12
1 @ 2x5 socket header @ J1 (OEM) or J2 (USB) or J3 (SD Card)
1 @ ALFAT daughterboard adapter (OEM or USB or SD Card)
Standoff hardware @ 12mm Standoff + 2 screws (2 sets for USB/SD, 4 sets for OEM)

Appendix J
Remote Terminal

EEPROM v2.02 created the Remote Terminal process, which allows the SmallZ80 to be operated in either 'Local' mode, or a combined 'Local+Remote' mode. The 'Local' mode is the normal connection: one terminal connected to one computer, somewhere nearby. The use of the LAN adapter means 'nearby' might be fairly distant, but the concept is still 'local'.

A Remote terminal, for our purposes, is one connected to the SmallZ80 through the Internet, and it can be halfway around the world.

The combination of 'Local+Remote' terminals means both terminals are connected at the same time, both are capable of entering keyboard commands, and both display the same data at all times. Obviously, to accomplish this, one of the various LAN Expansion adapters must be installed in the SmallZ80. The only other requirement is that the EEPROM and BIOS versions must be v2.02 or higher.

If the EEPROM initialization code detects that an expansion adapter is present, it will check to see if J14 is also installed. If both are true, then the 'ReRoute' mode is enabled and all terminal I/O will be conducted through the Main (local) terminal AND the Alternate (remote) terminal. The status of the ReRoute flag is passed to the BIOS at Cold Boot along with the base address of the Remote serial port, so the BIOS will operate in the same manner.

There are a couple of items to be aware of prior to enabling Remote operation:

– The Remote operator has the same FULL & EQUAL control over the SmallZ80 as the local operator, so it would be very easy to erase the IDE drive, for instance. There is absolutely ZERO security over the connection.

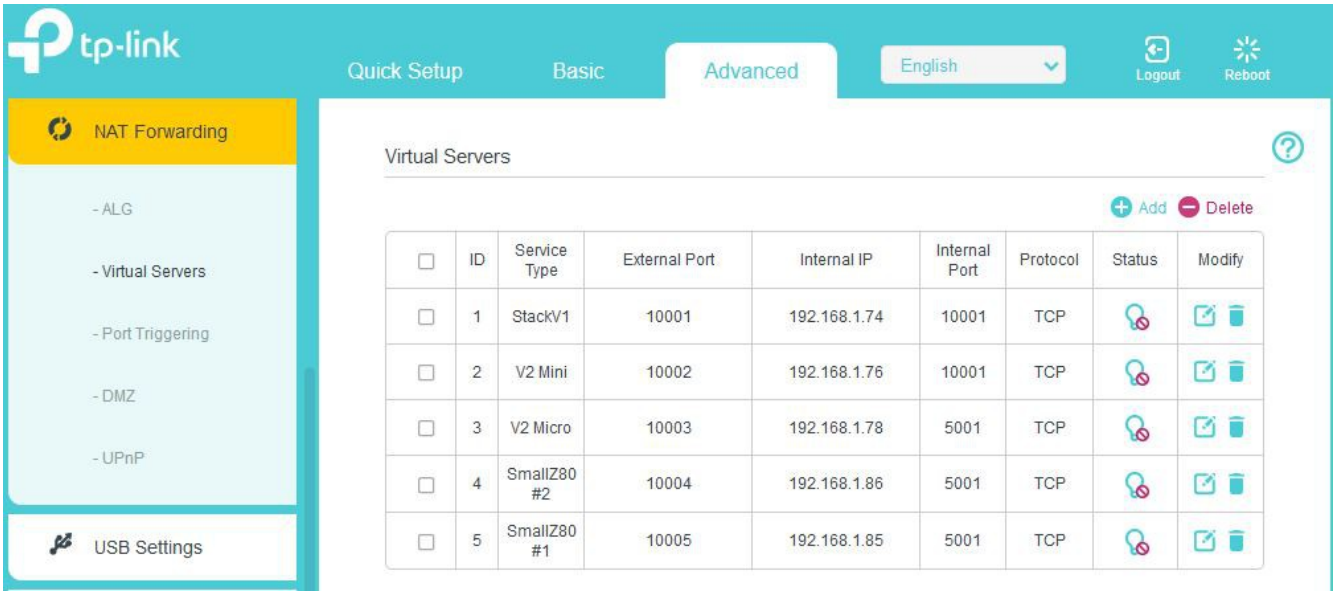
– The process of displaying everything twice does slow down the machine. You might only see it once, but it's being sent though two serial ports.

Your Modem or Router will determine the path you have to follow to connect your SmallZ80 to the Internet. Newer Modems can do it via the DMZ options; our Modem doesn't do it so we cannot advise you on it. Our Router allows VPN connections that DO allow us to specify an IP address on our network that can be 'attached' to a specific port. For instance, using our ISP-provided WAN IP address, along with a user-specified port, we can direct the data stream directly to a specific local IP address on our network.

For example, assume your WAN IP address is 45.67.3.200, and your local network IP address is 192.168.0.xxx. Further, assume your SmallZ80 local network IP address is 192.168.0.100, and uses port 5001 for terminal I/O. Internet access to your SmallZ80 would be at IP Address 45.67.3.200, and the port would be specified as part of your router setup. We use port 10001, so the full IP address would be 45.67.3.200:10001. The router would then connect that incoming data stream to the SmallZ80 at 192.168.0.100:5001.

Appendix J
Remote Terminal (cont)

Here's a screenshot of our Router VPN table:

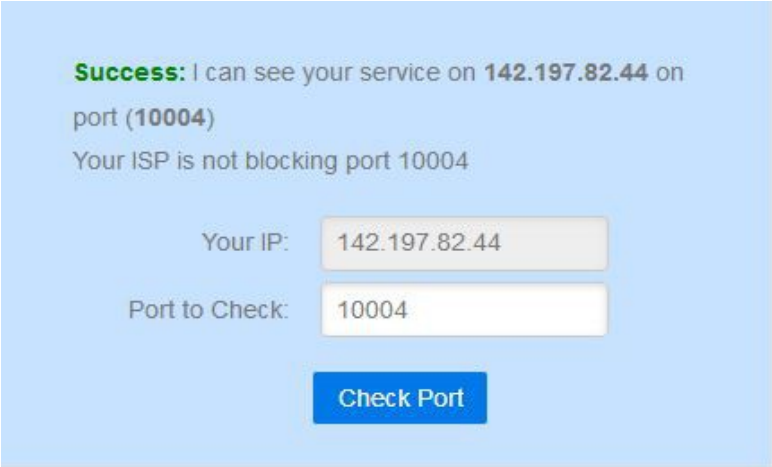


	ID	Service Type	External Port	Internal IP	Internal Port	Protocol	Status	Modify
☐	1	StackV1	10001	192.168.1.74	10001	TCP		
☐	2	V2 Mini	10002	192.168.1.76	10001	TCP		
☐	3	V2 Micro	10003	192.168.1.78	5001	TCP		
☐	4	SmallZ80 #2	10004	192.168.1.86	5001	TCP		
☐	5	SmallZ80 #1	10005	192.168.1.85	5001	TCP		

As you can see, the Internet will 'see' the SmallZ80 #1 at our WAN IP address, port 10005. The Router will re-direct any traffic at that port to the SmallZ80 #1, at 192.168.1.85:5001.

Usually, your Modem or Router can display the WAN IP address that is assigned to your Modem by your ISP. If not, you can use your browser to this website: https://canyouseeme.org/

Notice the 'slash circle' on the light bulbs in the Status column, signifying that the Virtual Server is disabled. Using the canyouseeme.org website to 'see' a disabled server won't work. When it does work, you'll see this:



Appendix J
Remote Terminal (cont)

Once you get the link established in the router, you'll need to have your SmallZ80 on-line and ready to accept commands. Since the modem/router is used to connect to the Internet, the SmallZ80 LAN port (on Serial-D) must be used for the Remote terminal, leaving one of the other three serial ports (A, B, or C) for use by the Local terminal. EEPROM Setup, in the System Defaults menu, Option 4 is used to set the ReRoute Base Port. It defaults to Serial-B, but can be any of A, B, or C.

It is not necessary, but if you wish to observe or interact with your Remote user, you can establish the Local connection. You will be able to see everything the remote user does, and you can enter commands, data, or chatty text at will. (Chatty text at the CP/M command line will cause a command error, but the error can be ignored. Unless your chatty text is something like ERASE *.*, in which case it's not chatty text – it's a command that will erase every file in the current directory – not normally something you want to do.)

One final note – your WAN IP address, provided by your ISP, is a DHCP-assigned address, and it can change at any time without notice. If you wish to establish a permanent IP address, you'll likely need to open a VPN account. On the other hand, our WAN IP address has been constant for several days, so it probably won't change very often.