Proton - Z80 Modular Computer
User's Manual
Version 1.0

Developed by Ricardo Kaltchuk
kaltchuk@gmail.com

PRELIMINARY VERSION

*** INCOMPLETE ***

UNDER CONSTRUCTION

Preface

Two years ago I was searching for some ICs in my component drawers and came across a Z80 CPU and an Intel 8251 USART. I had an instant flashback and remembered my first contact with a microprocessor back in 1982. It was a Z80. After a few minutes of nostalgia I decided to make good use of these components and build a computer.

Of course I couldn't avoid thinking about this "new project" with my Systems Engineer mindset. So, I established a few initial goals:

- This project is a hobby, so it has to be fun.
- It should use mainly technologies available in the 1980's.
- The computer has to run commercial software.
- The architecture should be modular allowing for new cards with specific functions.
- It also had to be a hardware/software development platform for students, hobbyists and enthusiasts.

After some research I decided to design the hardware fully compatible with CP/M 2.2.

One and a half years later I finally had my wire wrapped computer running CP/M. I could write programs using WordStar and compile them with a BDS C compiler or Microsoft Basic compiler. It was fun, exciting and challenging. After that, I decided it was time to turn the prototype into a finished product.

It's gratifying to see that in two years, alone, I was able to transform an idea into a final product.

Ricardo Kaltchuk
Ashdod, october 2021.

"Anyone who has brought up CP/M on a homebuilt computer has felt this moment of elation. A myriad of connections are properly closed; bits are flying at lightning speeds over buses and through circuits and program logic, to produce a single prompt".

Gary Kildall

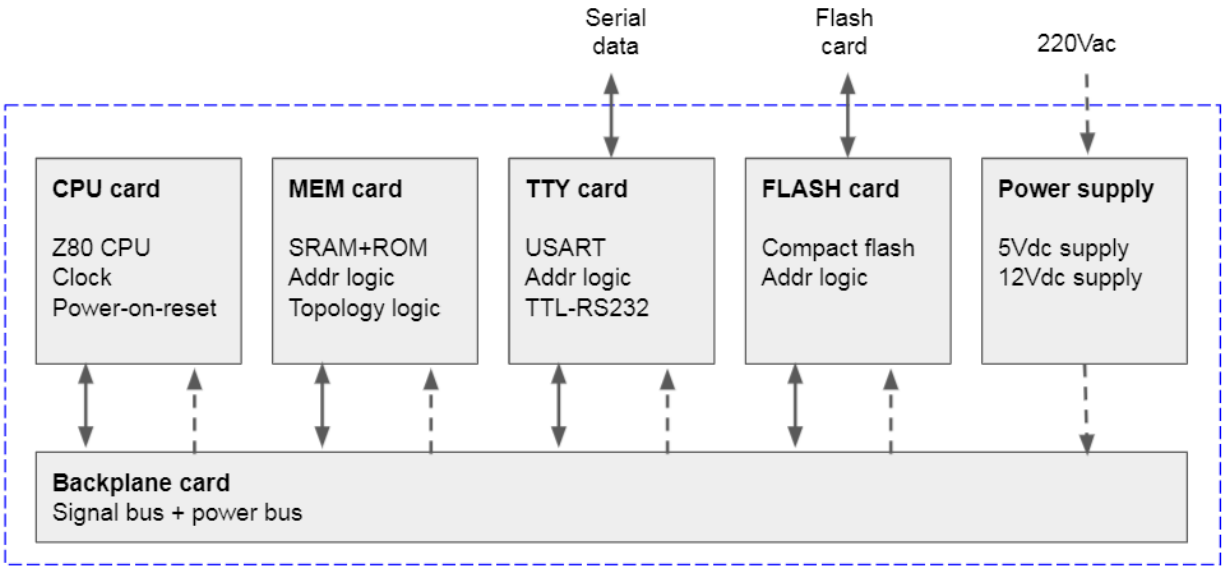
Index

Introduction

The Proton Z80 Modular Computer (Z80MC for short) has a minimal configuration of six modules:

1. CPU Card - Z80 CPU, clock circuit and power-on-reset.
2. MEM Card - 64KB of ROM and 128KB of RAM.
3. FLASH card - 128MB of compact flash emulating 16 logical drives.
4. TTY Card - two serial ports, RS232 or TTL level.
5. PSU Card - 5V power supply.
6. Backplane card - passive PCB that connects all cards.

Every card that exchanges data with the CPU, has a unique hexadecimal 4-bit I/O address ranging from 0 to Fh. The addresses are configured via jumpers on each card. Even the MEM Card has an I/O address although it's not an I/O peripheral *per se* (more details on the MEM Card section). All the modules described above are assembled in a rack. The rack may contain more than one unit of each module, except for the CPU Card, MEM Card and PSU Card (5Vdc).



Proton Z80 Modular Computer system diagram.

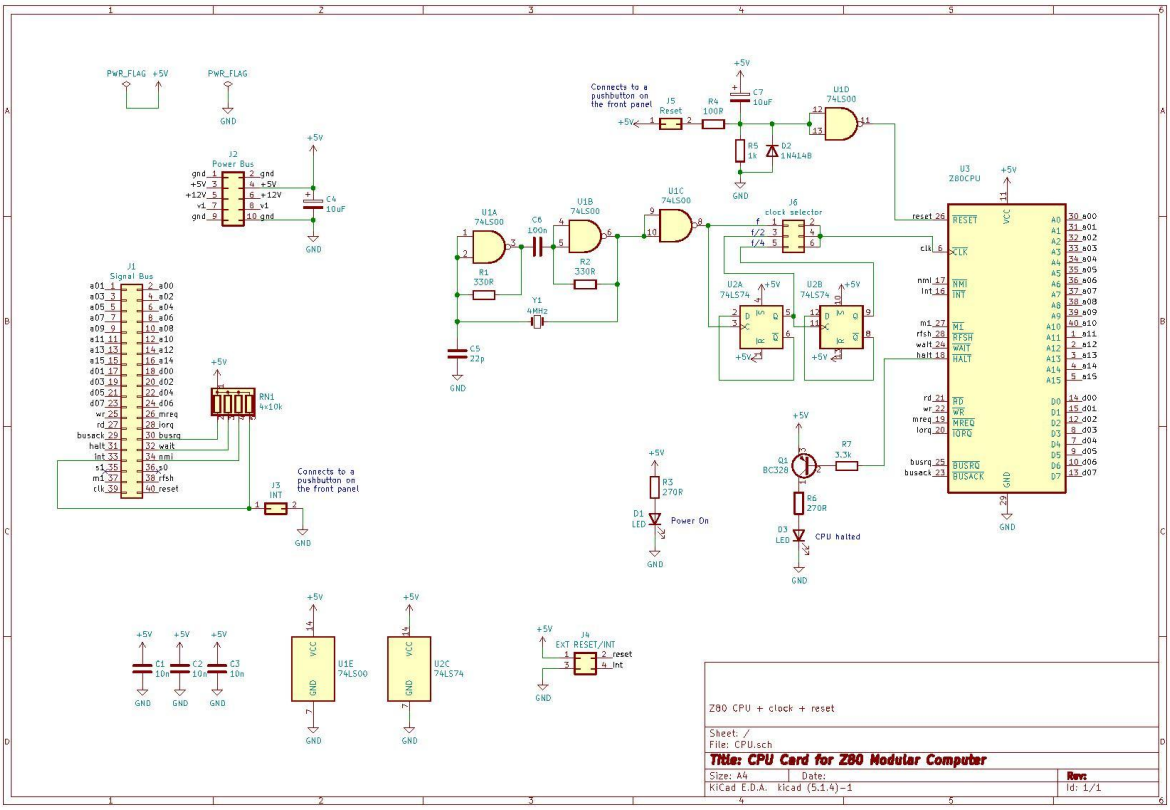
Please read all this manual before assembling, configuring and testing the Proton Z80 Modular Computer.

CPU Card

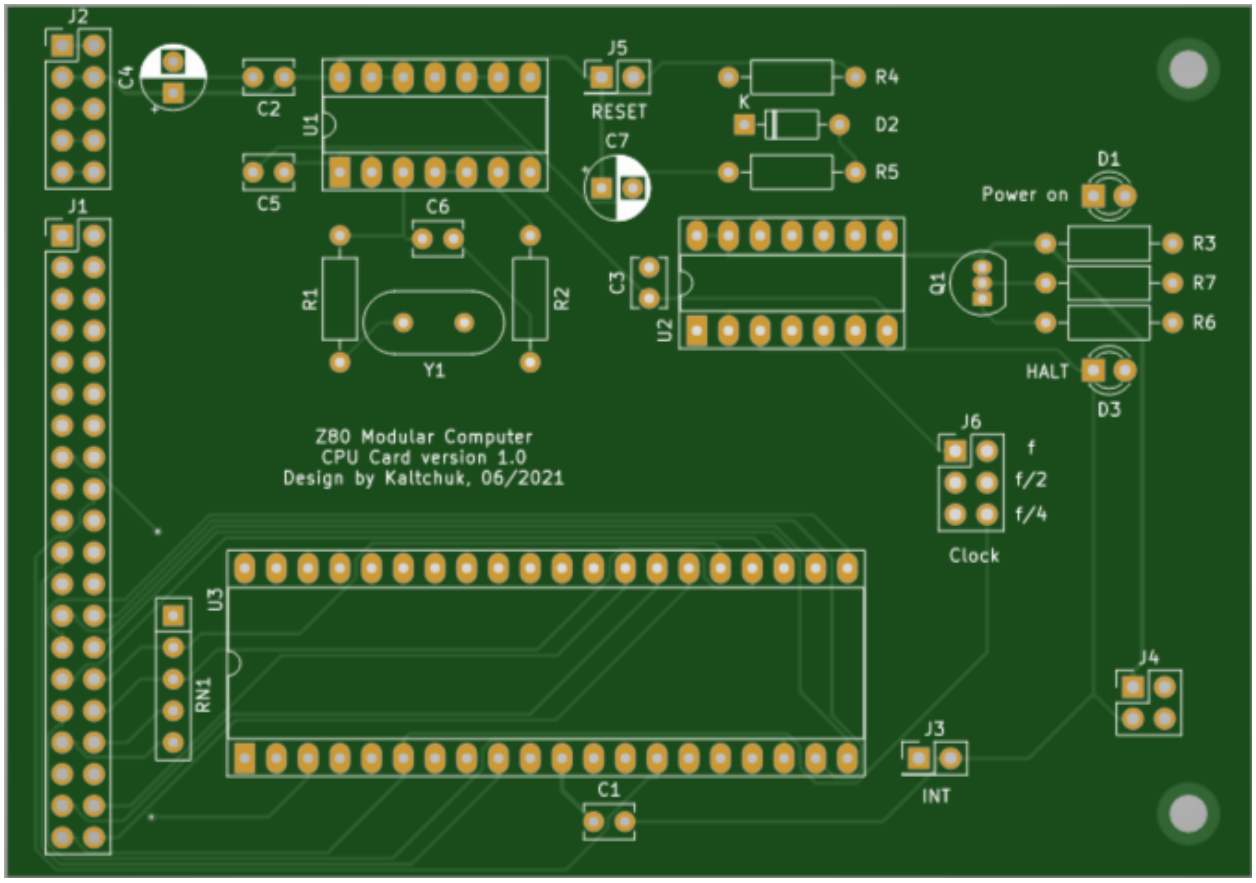
Description

The CPU Card has the Z80 CPU, the clock circuit and a power-on-reset circuit. The clock circuit uses an 8MHz crystal and two flip-flops to divide the frequency (4 and 2MHz).

The card has a green LED to indicate power-on and a yellow LED that shows when the CPU is in HALT state.To reset card, press the reset button. This will hold the CPUs reset pin down for a few microseconds. The reset is automatically performed during power-on.



CPU Card schematic.



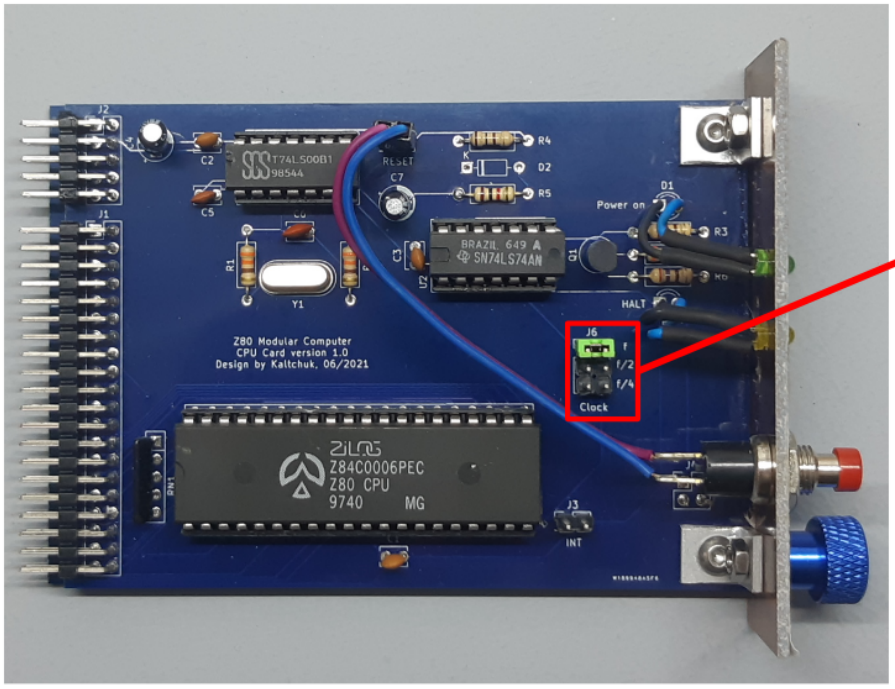
CPU Card PCB.

CPU Card Frontal

Bill of Material

Configuration

Clock select (J6). The only configuration on the CPU Card is the clock frequency. According to the position of jumper J6, the CPU can run at 3 different speeds: f, f/2 or f/4; where f is the cristal's frequency. Since the original project uses an 8MHz crystal, the three possible speeds are: 8, 4 or 2 MHz.



CPU Card

J6: Clock Select

MEM Card

Description

The MEM Card has 64KB of EEPROM and 128KB of SRAM. The CPU can only access 64KB of memory, so there are some circuits that handle the memory topology.

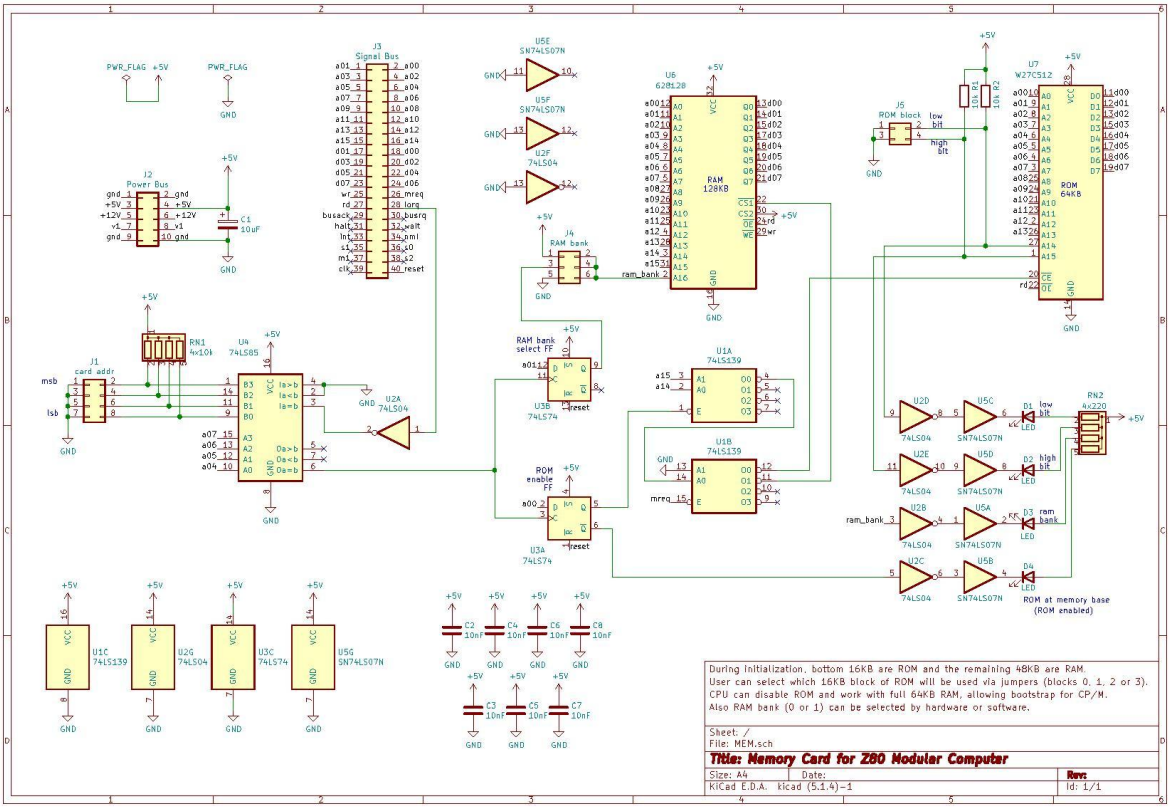
The EEPROM is divided into 4 blocks of 16KB - blocks 0, 1, 2 and 3. Block 0 contains CP/M 2.2 operating system and the BIOS. Block 1 contains Monitor 2.0 and the BIOS. Blocks 2 and 3 are empty and can be programmed by the user for other needs. The SRAM is divided into 2 banks of 64KB - banks 0 and 1.

When the Z80MC is powered up and also after reset, the lower 16KB of memory (0000 - 3FFF) are allocated to one of the EEPROM blocks. The upper 48KB of memory (4000 - FFFF) is allocated to one of the SRAM banks. The bootloader on the EEPROM copies CP/M and the BIOS (or Monitor and the BIOS if block 1 is selected) to the top of the SRAM (D000 - FFFF) and disables the EEPROM, transforming all 64KB in SRAM.

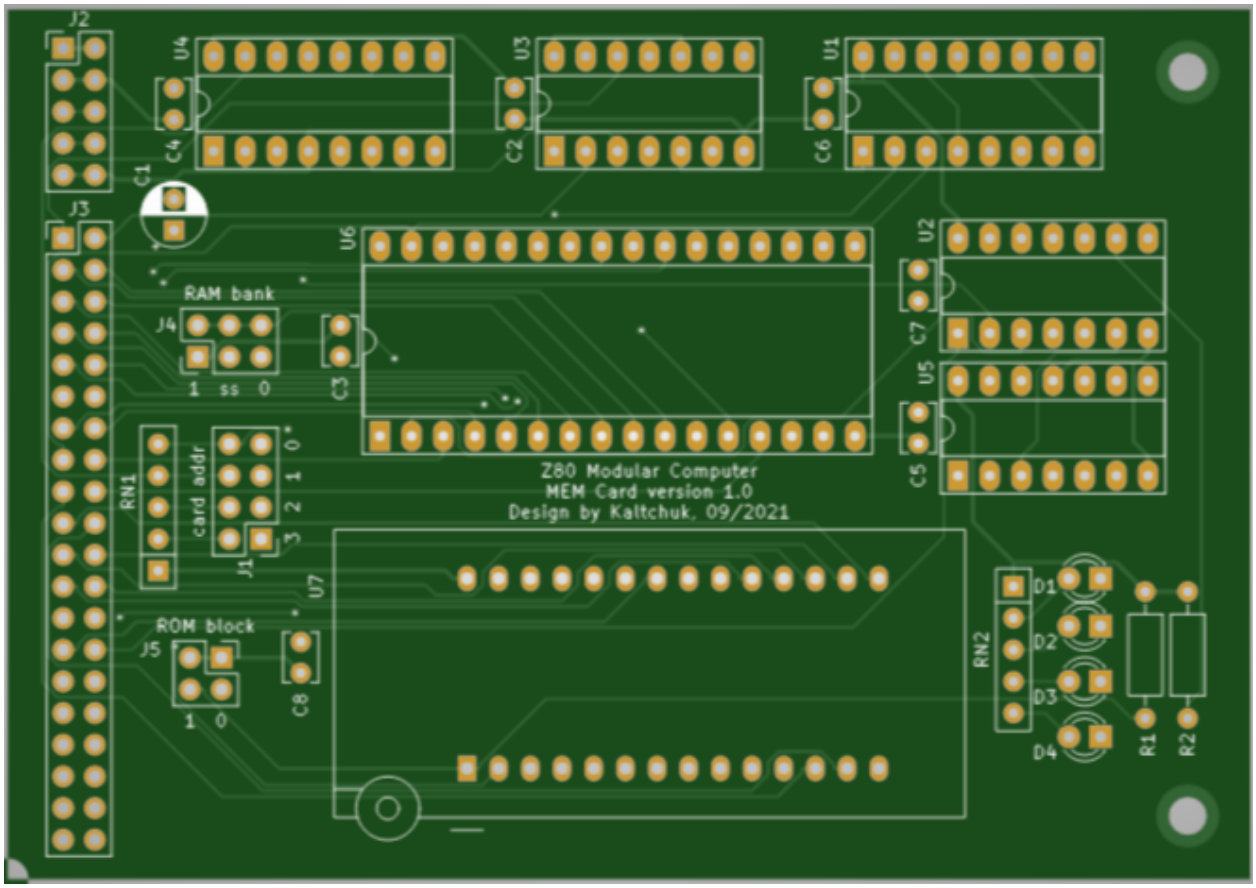
The frontal has two LEDs to indicate which ROM block is selected and one LED to indicate if the ROM is active, i.e. if the bottom 16KB is being used by the ROM. As soon as the BIOS starts executing it disables the ROM and the LED will turn off.

There is also one LED that indicates which RAM bank is being used.

As the Proton Z80MC is intended for experimentation, the EEPROM is placed on a ZIF socket to ease insertion/removal for reprogramming.



MEM Card schematic.



MEM Card PCB.

MEM Card Frontal

Bill of Material

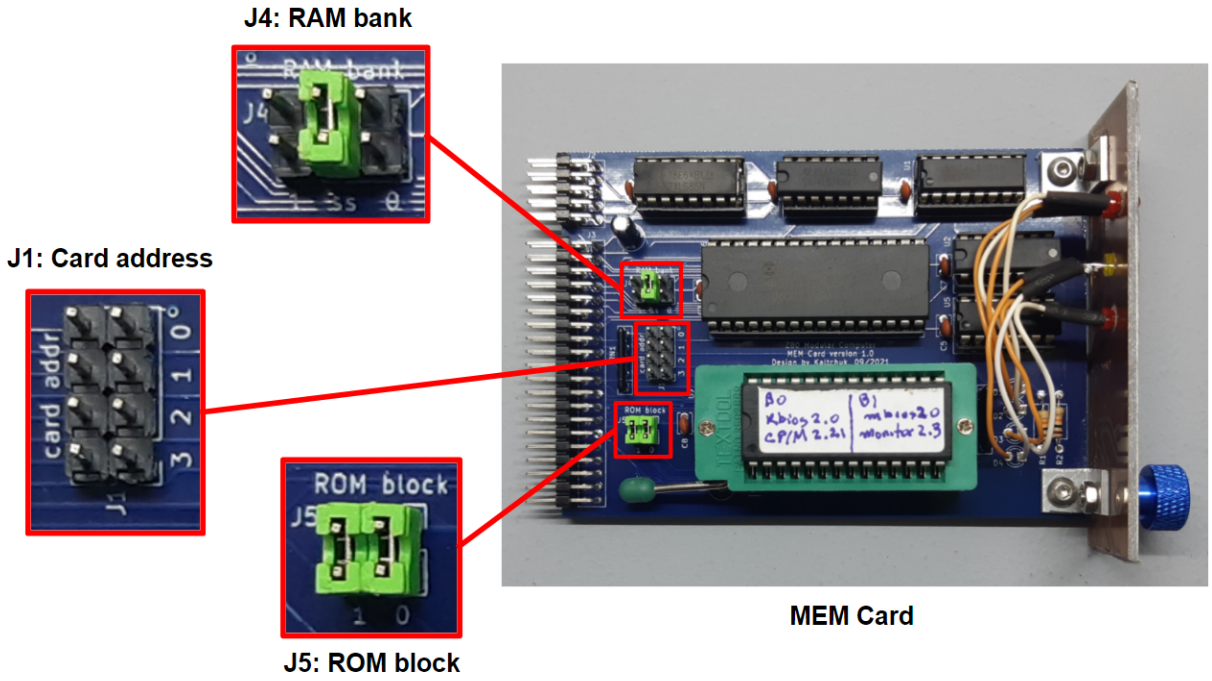
Configuration

There are three configurations needed on the MEM Card.

Card address (J1). This is the address that the CPU will use to communicate with this card via IN/OUT instructions. It is a byte from 0 to F. No jumper means 1 and jumper means 0. The picture shows the card configured with address F.

RAM bank (J4). Jumper 4 selects between bank 0, bank 1 or software selectable (ss) by the CPU.

ROM block (J5). Jumper 5 selects which block will be used to boot. No jumper means 1 and jumper means 0. The picture shows the card configured to boot from block 0..



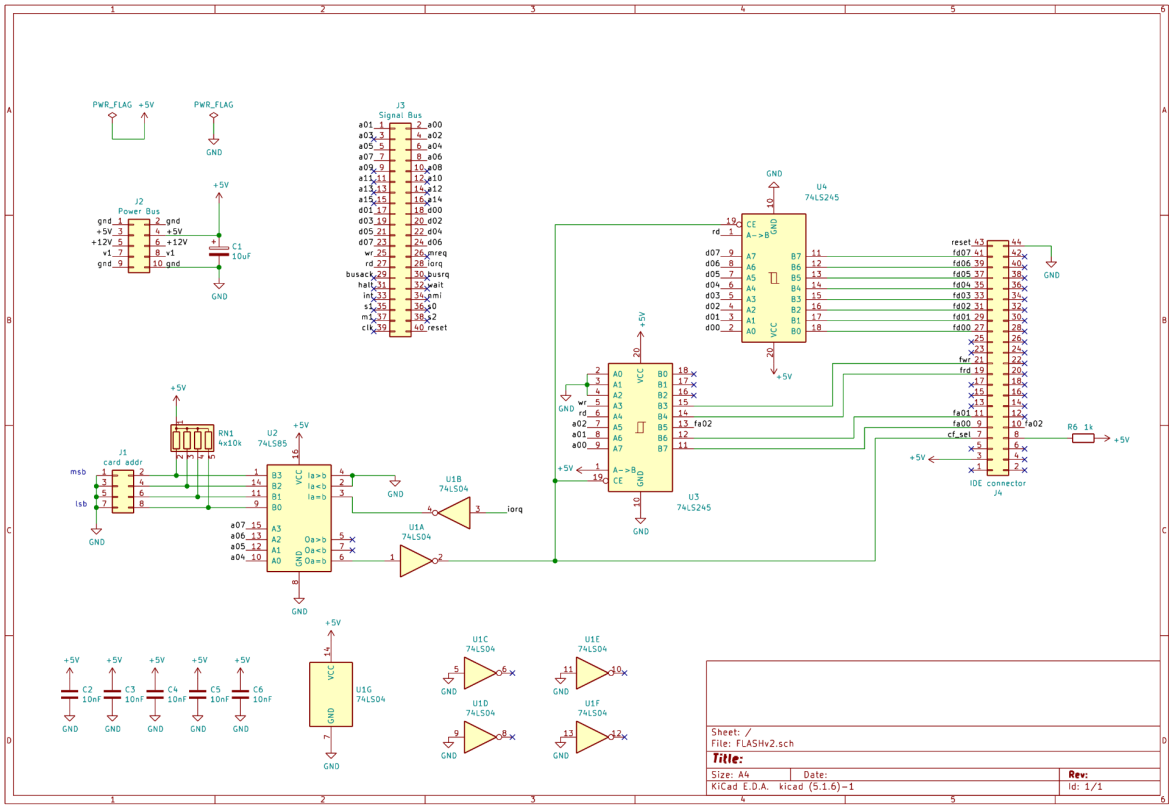
FLASH Card

Description

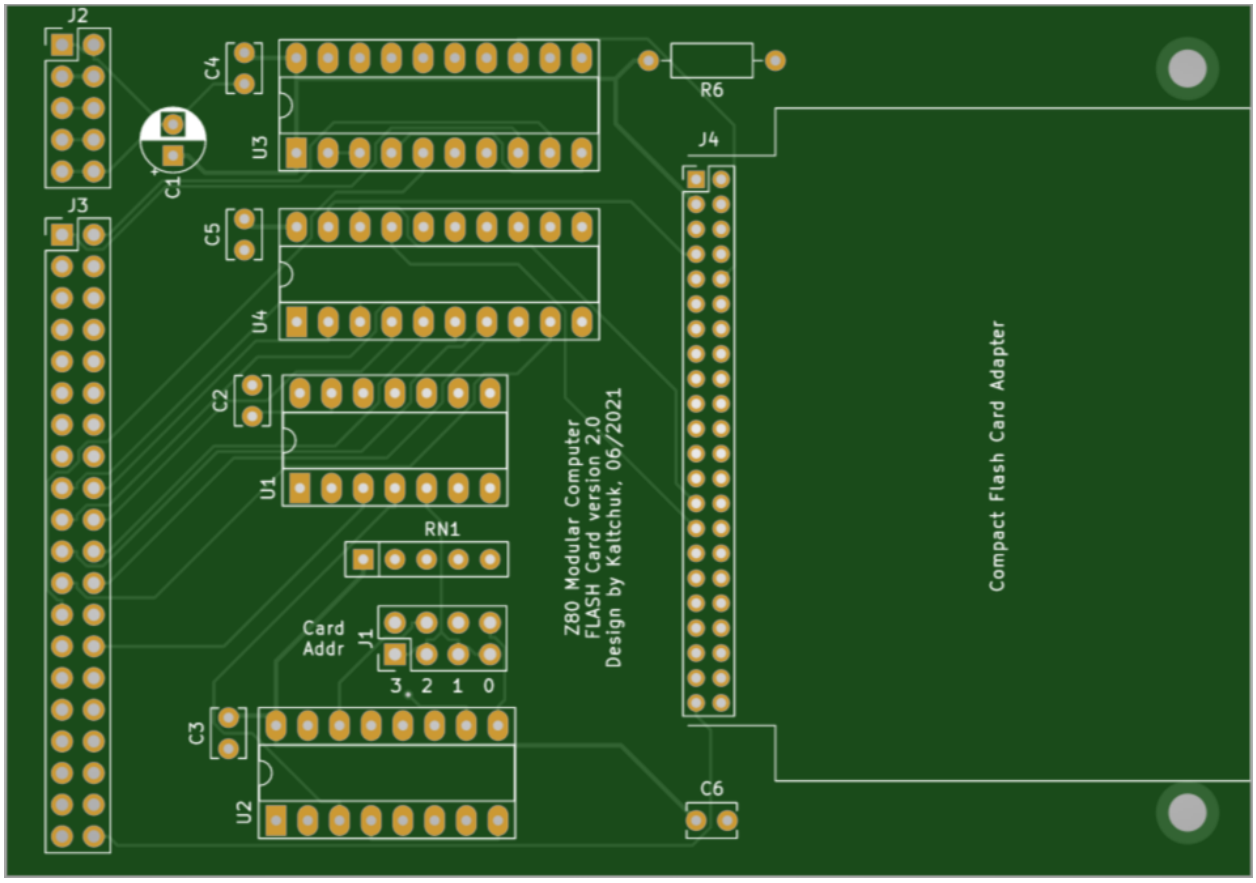
Since CP/M requires at least one floppy or hard drive to work, the FLASH Card emulates sixteen drives, each one containing 16MB. The compact flash is already formatted for use with CP/M and comes with several softwares and utilities. Drive "a" contains a text file named DISK.MAP that shows the summary of each drive's content.

The compact flash card seems to be very susceptible to noise on the signal lines, so version 2 of FLASH Card uses 74LS245 (octal bus transceiver with 3-state outputs) to improve the signal quality.

The frontal has an LED that indicates flash card activity (read/write).



FLASH Card schematic.



FLASH Card PCB.

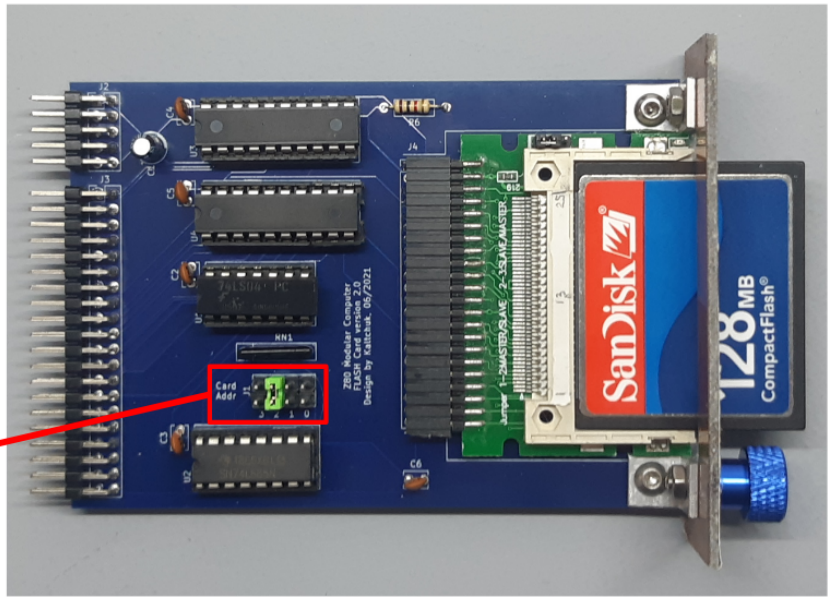
FLASH Card Frontal

Bill of Material

Configuration

Card address (J1). The only configuration on the FLASH Card is the card's address. This is the address that the CPU will use to communicate with this card via IN/OUT instructions. It is a byte from 0 to F. No jumper means 1 and jumper means 0. The picture shows the card configured with address B.

J1: Card address



FLASH Card

TTY Card

Description

The first attempt at a serial communication card was called USART Card and was implemented via an Intel 8251 USART. It had only one serial port and a maximum baud rate of 38400bps. It used an interrupt to signal the CPU that a byte had arrived. This interrupt also generated a conflict and did not allow some softwares like SID and ZSID to run. I decided it was time to try something else. Mainly, I had two options: a) use the Zilog SIO or, b) use a non-orthodox approach. When I thought of using a microcontroller I new that it would violate one of the initial goals - “use mainly technologies available in the 1980's”; but, on the other hand, there is another goal that said: “It also had to be a hardware/software development platform for students, hobbyists and enthusiasts”. Based on this, I decided to develop a card with two serial ports, each one controlled by an independent microcontroller; and based on my previous experience with Arduino, I chose the ATmega328.

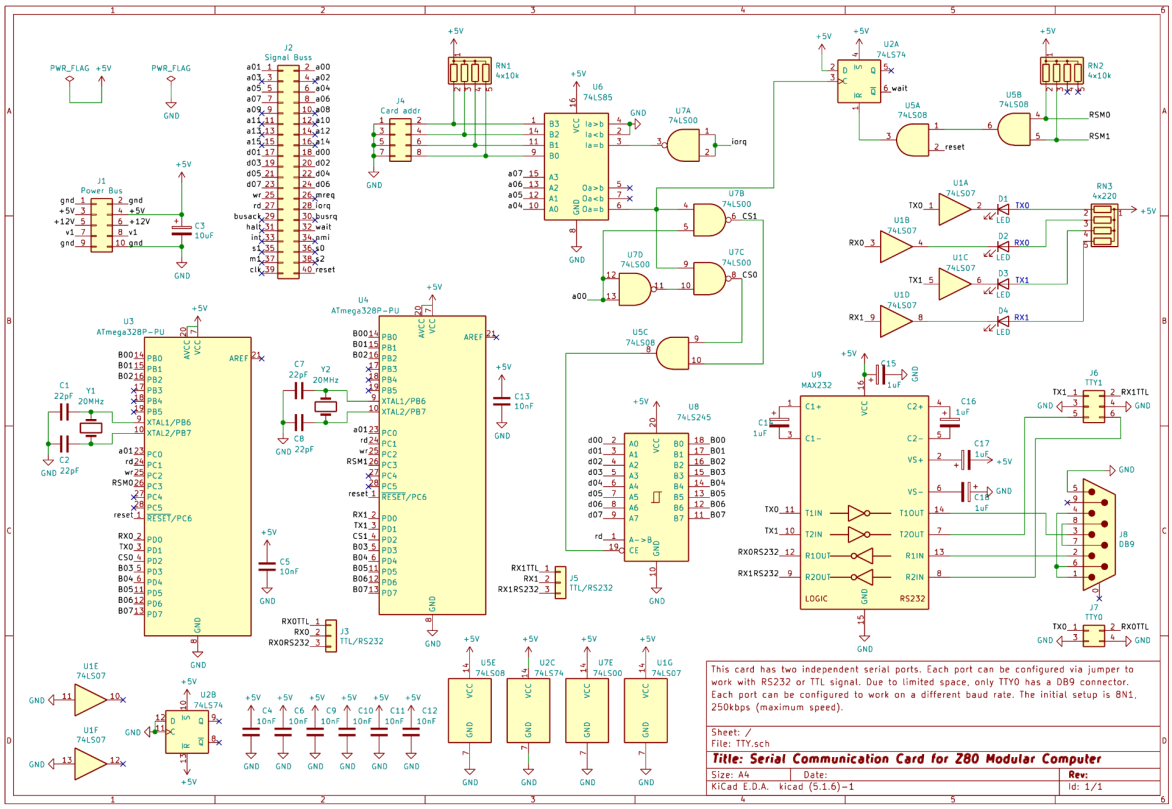
The TTY Card provides the primary means for the Z80MC to communicate with the user and other serial peripherals. It has two independent serial ports - TTY0 and TTY1. Each port can be configured to work at 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, 125000 or 250000 bps via software on CP/M. Run TTYBAUD.COM; select the port and the desired baud rate. After reset, both ports will start at 250kbps, 8 bits, no parity and 1 stop bit. Besides the baud rate, all parameters can be changed but this requires modifying the firmware on the ATmega328 microcontrollers.

Both ports can be configured by hardware to use RS232 or TTL level (see Configuration below). Due to limited space, only TTY0 has a DB9 connector.

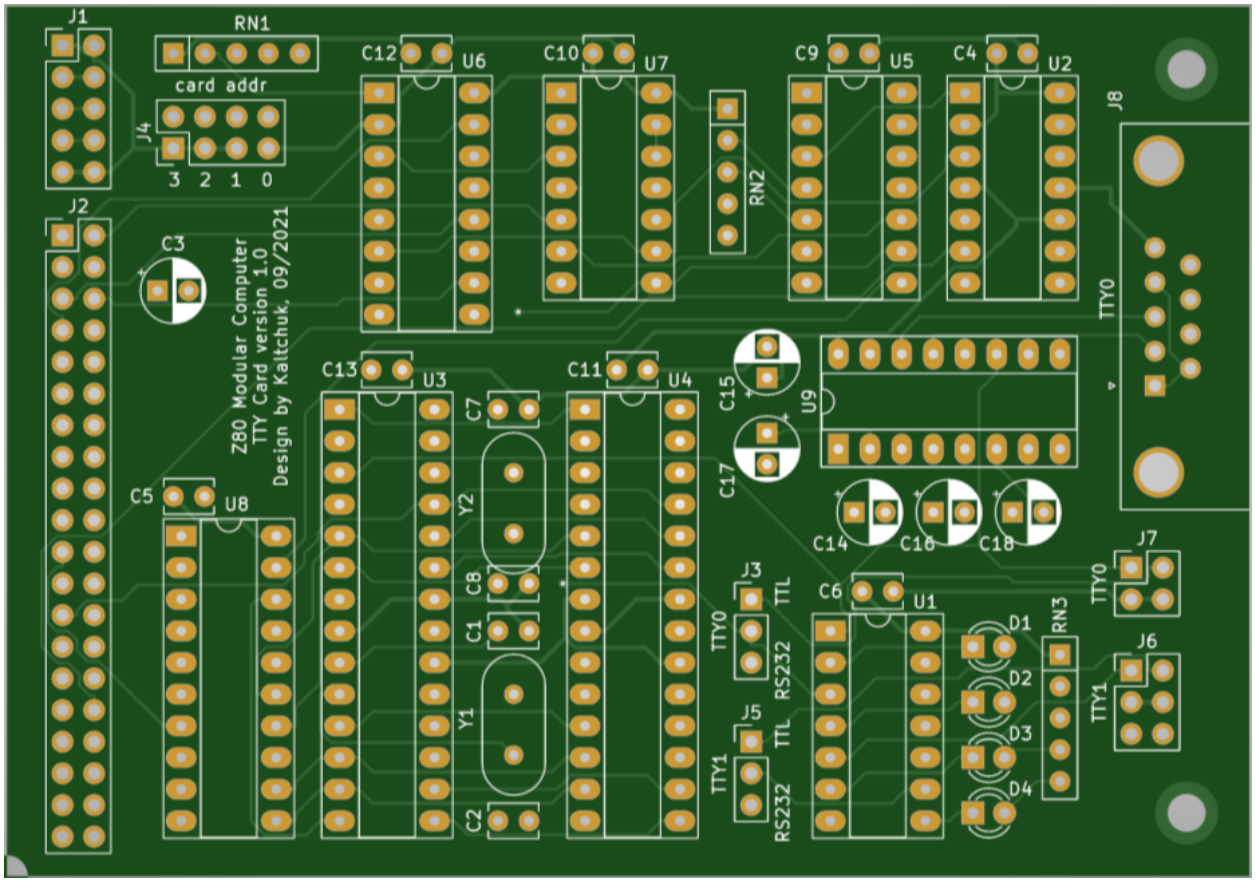
Each port has a 128 byte circular input buffer, which means that data will be lost if the CPU does not fetch the data before the buffer is full. The card does not inform the CPU when new data has arrived; so it is the operating system's responsibility to check for new data in the buffers.

The frontal has 4 LEDs, two for each port, the green LEDs are for RX and the blue LEDs are for TX.

The first address of the card is used by TTY0 and the second by TTY1. For example, let's say that the card is configured with address “C”. In this case, the CPU will communicate with TTY0 using input/output on address C0 and with TTY1 on address C1.



TTY Card schematic.



TTY Card PCB.

TTY Card Frontal

Bill of Material

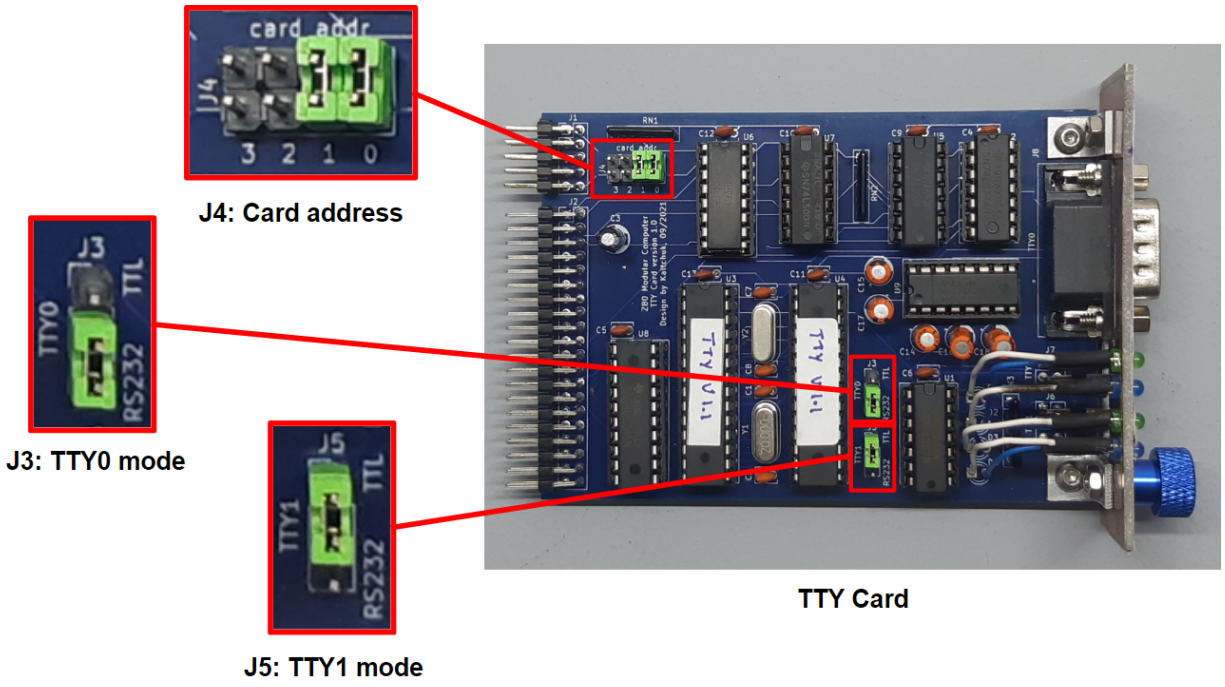
Configuration

There are three configurations needed on the MEM Card.

Card address (J4). This is the address that the CPU will use to communicate with this card via IN/OUT instructions. It is a byte from 0 to F. No jumper means 1 and jumper means 0. The picture shows the card configured with address F.

TTY0 mode (J3). This jumper selects if port TTY0 will communicate using RS232 or TTL level. The picture shows TTY0 configured as RS232.

TTY1 mode (J5). This jumper selects if port TTY1 will communicate using RS232 or TTL level. The picture shows TTY1 configured asTTL.

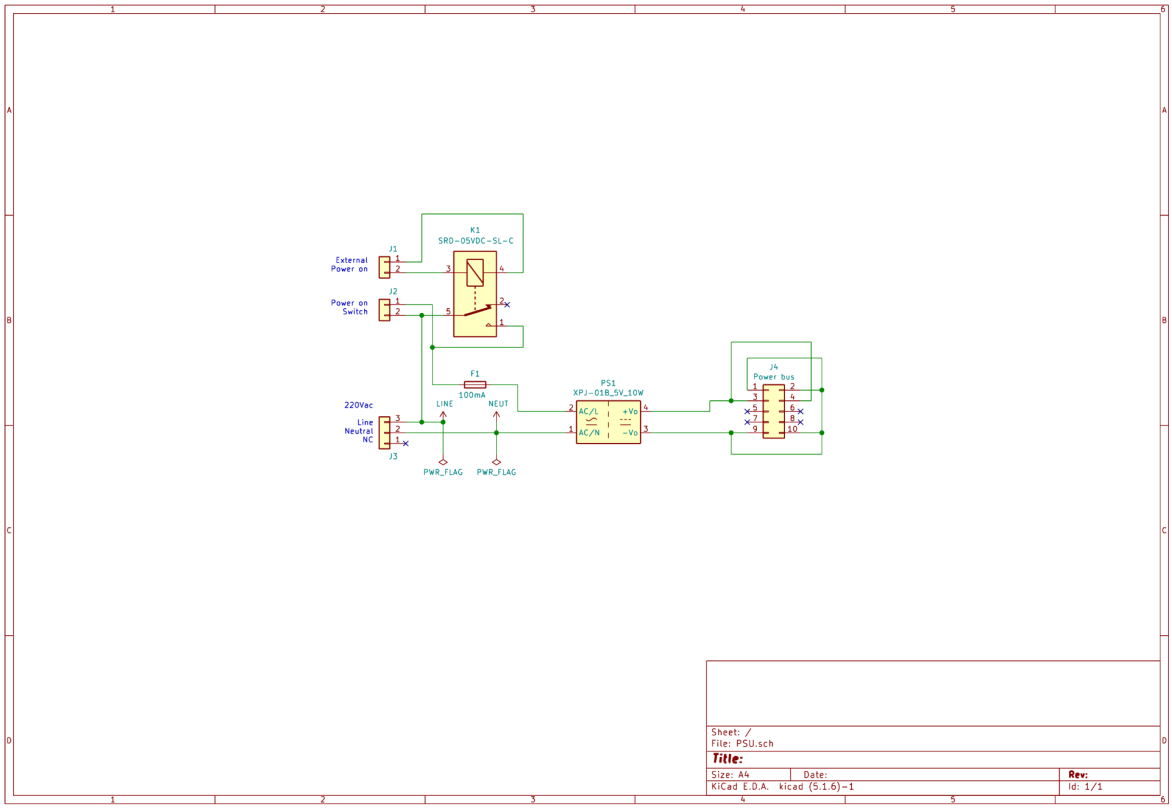


PSU Card

Description

The PSU Card provides the 5Vdc necessary for all other cards. It has a maximum output of 2A and has to be connected to an external 110 - 220Vac, 50/60Hz supply. The card has a 100mA fuse and an input for external power-on. For example, a modem can be used to turn on the PSU using a 5V signal.

The frontal has an ON/OFF switch, a connector for the power cord and a connector for external power-on command.



PSU Card schematic.

Figure - PCB.

Figure - Frontal.

Bill of Material

Configuration

No configuration is required for the PSU Card.

Backplane

Description

The Backplane Card is completely passive and has no electronic component, only connectors. Each Backplane card has slots for 3 modules and cascading connectors so several backplane cards can be interconnected. In order to run CP/M, a minimum of five modules are required - CPU, MEM, TTY FLASH and PSU, so at least two backplane cards must be used. This gives a total space for 6 modules (five slots will be used and one will be empty).

Backplane Schematic

Backplane PCB.

Bill of Material

Configuration

No configuration is required for the Backplane.

Protoboard

Description

The Protoboard is intended for experimentation and development of prototype modules for the Proton Z80MC. It was especially developed for prototyping using wire wrapping techniques.

All the signals from the signal and power buses can easily be accessed via the male pin headers.

Protoboard schematic.

Protoboard PCB.

Bill of Material

Configuration

No configuration is required for the Protoboard.

LCD Card (ONLY IN PROTOBOARD)

Description

The LCD Card was initially developed as an easy output during the debugging phase of the USART Card. It can be used as a pseudo printer.

Before using this card, a driver has to be installed. The easiest way to do this is by using the DRIVER.COM utility on drive A.

Building Procedure

Follow these steps to build your own **Proton Z80 Modular Computer**:

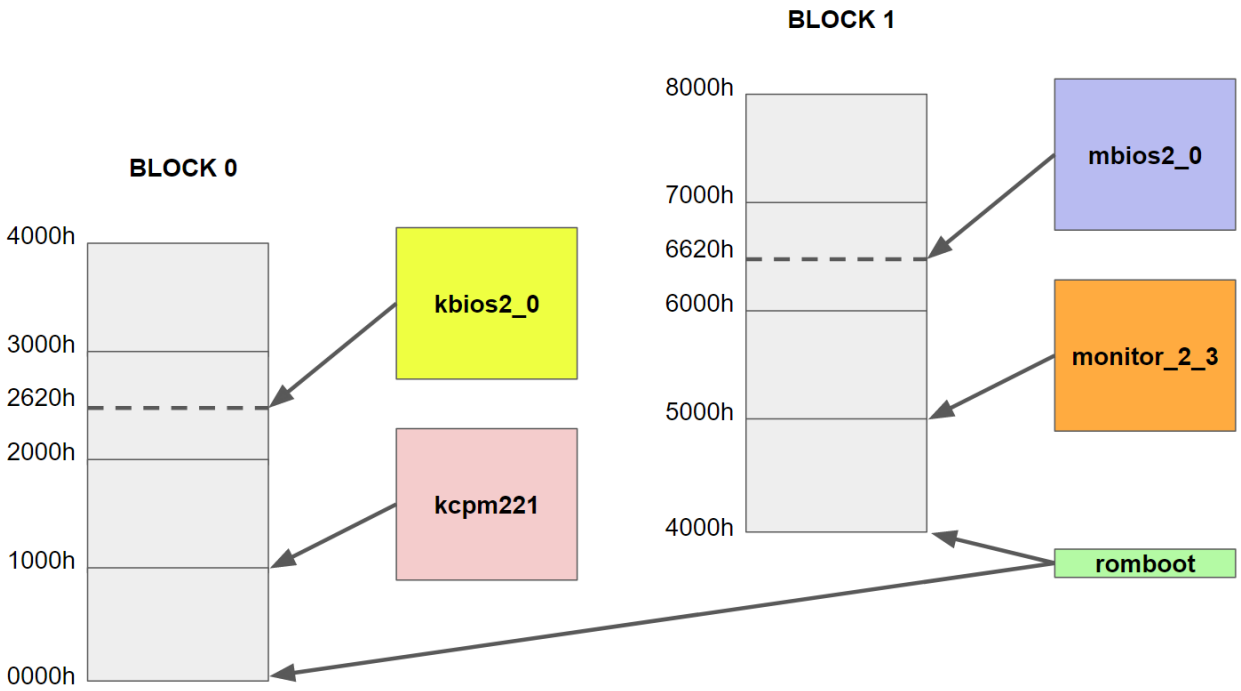
1. **PCB cards.** You can simply order the PCB cards from PCBway and JLC PCB. Check the following chapters for info about the PCBs.

2. **Components.** Each card has its own bill of material. Check the following chapters.

3. **Cards Assembly.** After you gather all the PCBs and components, it's time to assemble the cards. Check the pictures in Github to see what the cards should look like.

4. **Enclosure.** This is the most difficult part because there are no “final parts” for sale. You’ll have to cut and drill all the metal sheets and profiles. Check the Github for the mechanical sketches. The enclosure is not essential to get the Proton running, but it will surely give a professional look & feel.

5. **EEPROM.** You have to burn the EEPROM according to the following memory topology:



kbios2_0.obj, kcpm221.obj and romboot.obj are here:
https://github.com/KALTCHUK/Z80_Modular_Computer/tree/master/CPM%20build

mbios2_0.obj, monitor_2_3.obj and romboot.obj are here: https://github.com/KALTCHUK/Z80_Modular_Computer/tree/master/Monitor%20build/Monitor%202

6. **Microcontrollers.** At least one of the ATmega328 on the TTY Card has to be programmed. Both, if you intend to use also the second port (TTY1). You can find the program under Atmel Studio directory (https://github.com/KALTCHUK/Z80_Modular_Computer/tree/master/Atmel%20Studio/preTTY/preTTY). I used Atmel Studio 7 with an AVRISP mkII programmer.

7. **Configuration.** Each card, except the PSU, has a few jumpers that have to be configured. In the chapter for each card you'll find a section called "Configuration". It's important to notice that the address of each card is "hard coded" in the BIOS. So if you intend to use different addresses, you'll need to modify the BIOS.

   At this point, all the hardware is ready and configured. Insert all the cards in the rack, insert a 128MB compact flash in the FLASH Card, connect the power cord to the mains; and connect port TTY0 (DB9 connector) to your computer, via a USB-RS232 converter. (I suppose your computer doesn't have a native RS232 port). On your computer, open the serial terminal emulator and configure the port to 8N1, 250kbps. I strongly recommend the use of Teraterm as your serial terminal emulator. Select ROM block 1 (Monitor) and turn on the Proton. You should see the initial message from the TTY Card:

   ```
   *** TTY Card - firmware v1.1. ***
   *** by Kaltchuk, sep/2021.    ***
   ```

   After that, comes the boot message from Monitor:

   ```
   Z80 Modular Computer by Kaltchuk 2020.
   MBIOS 2.1
   Monitor by Kaltchuk 2020.
   ```

   Read the Monitor User's Manual and start exploring the Proton.

   Now, you can try to boot CP/M. Select ROM block 0 (CP/M) and turn on the Proton. You should see the same initial message from the TTY Card as before, and the boot message from CP/M:

   ```
   Z80 Modular Computer by Kaltchuk 2020.
   KBIOS 2.0 - 128MB Compact Flash.
   CP/M 2.2 Copyright 1979 (c) by Digital Research
   ```

8. **Disk drives.** In order to use the Flash Card emulating floppy disk drives on CP/M you have to format all 16 logical drives on the flash card. Use the Monitor for this task.

Boot from block 1 (Monitor) and execute the format command for each one of the 16 drives (A, B, C… P).

```
>format a
```

To check if everything went ok, boot from block 0 (CP/M) and use the DIR command to view the content of each drive.

```
A>dir
```

All 16 drives should be empty.

9. **Software.** So far, so good. Now it's time to download the software, but first, we need some tools. The first tool we need is XMODEM.COM. Boot from Monitor and type:

```
>xmodem r 0100
```

Send XMODEM.COM, which can be found here: https://github.com/KALTCHUK/Z80_Modular_Computer/tree/master/CPM%20software/Backups/tools (Of course, I'm assuming you already downloaded XMODEM.COM from Github to your computer before you attempt to send it to Proton with xmodem).

After the transmission is complete, switch to ROM block 0 (CP/M), WITHOUT TURNING OFF THE PROTON. This procedure will swap from Monitor to CP/M without losing the file you just downloaded. Press the reset button and type:

```
A>save 4 modem.com
```

Type **xmodem** and you should get this message:

```
A>xmodem
MISSING FILE NAME

XMODEM 1.3 (10MHz build) - by Kaltchuk, 2021.
Use: XMODEM [drive:]file_name -<option>
options:    R to receive file
            S to send file
(Ctrl-X to abort transmission)
```

Now we have to download NULU.COM, which is the librarian, used to gather several files inside one .LBR file. This makes life much easier for up/download via xmodem. NULU.COM can be found in the same Github directory as XMODEM.COM. Type:

```
A>xmodem nulu.com -r
```

Send NULU.COM.

To test if NULU.COM downloaded correctly, type **nulu** and you should see this message:

```
P>nulu
NULU 1.52  (07/12/87)
Copyright (C) 1984, 1985 & 1987 by Martin Murray
Bug fixes in version 1.52 by Mick Waters

TYPE -H FOR HELP


NULU    .COM | XMODEM  .COM |
Drive P: Total 2032k, Used 20k, Free 2012k

-Open a library P0:>
```

Type **-x** to exit NULU.

At this point, you're ready to download all the software. As I mentioned before, the 128MB compact flash has 16 drives (a, b, c… p). Except for XMODEM.COM and NULU.COM, that you just downloaded to drive a:, all the rest is empty. The recommended content is shown in the table below:

```
           SUMMARY OF DISK CONTENT
           =======================

A:CP/M transient commands and utilities.
-----------------------------------------------------------------
B:MBASIC v5.21 - Microsoft Basic Interpreter, and Basic programs.
  BASCOM v?.?? - Microsoft Basic Compiler.
-----------------------------------------------------------------
C:BDS C Compiler v1.60 and C programs.
-----------------------------------------------------------------
D:SLR Z80 Assembler v1.30 and .Z80 files.
-----------------------------------------------------------------
E:WordStar 3.0
  TE 1.70 text editor
-----------------------------------------------------------------
F:FORTH v2.1.0
-----------------------------------------------------------------
G:Sorcim Supercalc v1.12 and spreadsheets
-----------------------------------------------------------------
```

```
H:dBASE II - Ashton-Tate Database Manager
----------------------------------------------------------------
I:
----------------------------------------------------------------
J:
----------------------------------------------------------------
K:
----------------------------------------------------------------
L:
----------------------------------------------------------------
M:area used for backup and restore
----------------------------------------------------------------
N:area used for backup and restore
----------------------------------------------------------------
O:
----------------------------------------------------------------
P:draft area
```

As you can see, only drives a: through h: are used. The easiest way to download all these programs is downloading the latest backup files from Github (https://github.com/KALTCHUK/Z80_Modular_Computer/tree/master/CPM%20software/Backups) with XMODEM and unpacking them with NULU. Let's do this procedure for drive a:.

Suppose that the latest backup from drive a: is A220128.LBR, which was produced on 28/01/22. You have to download it to drive n: and extract the files to the destination drive (a:). Type:

```
A>xmodem n:a220128.lbr -r
```

Send A220128.LBR. Type:

```
A>nulu -o n:l220128.lbr -e a:*.* -x
```

Now type **DIR** and you should get something like this:

```
A>dir
A: NULU     COM : HALT     COM : PROTON   COM : DELBR    COM
A: STAT     COM : PIP      COM : NULU     DOC : DDT      COM
A: DUMP     COM : ED       COM : LOAD     COM : USQ      COM
A: SUBMIT   COM : ZSID     COM : SCREEN   COM : BACKUP   DOC
A: DIRALL   SUB : SQ       COM : UNERA    COM : LU       COM
A: UNARC    COM : ANYDISK  COM : XMODEM   COM : LDIR     COM
A: UNCRUNCH COM : XDIR     COM : CRUNCH   COM : UNZIP    COM
A: LCDDRV   COM : ARK      COM : D        COM : NULU     INF
```

```
A: PK       COM : PK       DOC : HP       COM : HP       DOC
A: N41      COM : N41      DOC : N41      ASM : BACKUP   SUB
A: TTYBAUD  COM : DISK     MAP : RELEASE  NOT
```

Maybe the order is different, but the content should be very similar. Repeat the above procedure (xmodem and nulu) for drives b: through h:. Don't forget to change the destination drive in the **`nulu`** command line (**`-e <dest_drive>:*.*`**).

Congratulations, mission accomplished! Feel free to explore your new **Proton Z80 Modular Computer** with **CP/M 2.2** and have fun!

Please, send me an email with your feedback - kaltchuk@gmail.com.