

Z180 Computer

Contents

GitHub Repository

Z180 Computer description

Jumper configuration
Port functions

Board layout

KiCAD

ATF22V10 PLD

Design ideas, thoughts and questions

Z180 MMU
Clock, XTAL and Serial baudrate
CSI/O based SPI (or i2c) interface to SD card
AVR / Arduino controller
EPROM / FLASH jumper
Single step circuit (maybe TODO)

Test program

Components

References

GitHub Repository

2021-07-08

[hansake/Z180_Computer](https://github.com/hansake/Z180_Computer) (https://github.com/hansake/Z180_Computer)

Z180 Computer description

2021-07-28

Includes a Z180 CPU which has a simple MMU extending the 64KB logical address range to a physical address range of 1MB. This Z180 computer also includes 1MB of RAM memory and 256KB of EPROM memory. The lower 256KB physical memory is switchable between EPROM and RAM.

The memory in the EPROM socket may also be a FLASH memory: [S93WD662/WD663 DS - 39SF020.pdf](http://www.alphacron.de/download/hardware/39SF020.pdf) (<http://www.alphacron.de/download/hardware/39SF020.pdf>).

The logic for memory selection is implemented in an ATF22V10 PLD. PLD programmer: [hansake/PLD_programmer](https://github.com/hansake/PLD_programmer): GAL/PLD programmer used to program Atmel ATF22V10 and ATF16V8. (https://github.com/hansake/PLD_programmer).

A manual reset connector is available. The design is a bit peculiar as reset is made when the reset switch opens, this is to make use of the MCP130 supervisory circuit to create a stable reset when the supply voltage is out of bounds or the reset switch is pressed.

The interfaces include two RS-232 ports, two SPI interfaces to use with SD cards. A SD Card Adapter is needed as an interface between SPI and the SD cards.

- [How to use the "MicroSD Card Adapter" with the Arduino Uno | Michael Schoeffler](https://mschoeffer.com/2014/05/10/how-to-use-the-microsd-card-adapter-with-the-arduino-uno/) (<https://mschoeffer.com/2014/05/10/how-to-use-the-microsd-card-adapter-with-the-arduino-uno/>)

[effler.com/2017/02/22/how-to-use-the-microsd-card-adapter-with-the-arduino-uno/\)\).](http://effler.com/2017/02/22/how-to-use-the-microsd-card-adapter-with-the-arduino-uno/)

- SOLVED. Nrf24 (Mirf lib) + Micro SD-card works OK together - Using Arduino / Storage - Arduino Forum (<https://forum.arduino.cc/t/solved-nrf24-mirf-lib-micro-sd-card-works-ok-together/347787/9>)

An ATmega328P (the IC used in Arduino UNO) with SPI or Tx/Rx interfaces to the Z180 is also available. The connectors to the ATmega328P are intended to make it possible to use Arduino shields to add functionality to the Z180 computer. In the SPI connection between the Z180 CSI/O interface configured as a SPI master and the ATmega328P configured as a SPI slave.

The ATmega328P has a separate reset that is controlled by the Z180.

Using Z180 CSI/O as a SPI master seems possible according to: [SC126, v1.0, Circuit Explained | Small Computer Central \(ht
tps://smallcomputercentral.wordpress.com/sc126-z180-motherboard-rc2014/sc126-v1-0-circuit-explained/\)](https://smallcomputercentral.wordpress.com/sc126-z180-motherboard-rc2014/sc126-v1-0-circuit-explained/). Using ATmega328P as SPI slave is described in: [Serial peripheral interface in AVR microcontrollers - Embeddds \(https://embeddds.co
m/serial-peripheral-interface-in-avr-microcontrollers/\)](https://embeddds.com/serial-peripheral-interface-in-avr-microcontrollers/).

The ATmega328P Tx and Rx pins may be connected to the Z180 ASCI 1 Rx and Tx pins, or directly to one of the external RS-232 connectors via one of the MAX232 converters.

I am planning to program the ATmega328P with Pocket AVR Programmer Hookup Guide - learn.sparkfun.com/tutorials/pocket-avr-programmer-hookup-guide?_ga=2.127691909.94672799.1626256475-796128395.1619009331.

My selection of components is mainly based on what I found in the component boxes tucked away in my basement.

Jumper configuration

JP1 Tx Jumper

- Pin 1: Connected to channel 1 Tx D-sub output via RS-232
- Pin 2: Connected to channel 1 Tx from Z180
- Pin 3: Connected to Rx input on ATmega328P
- Pin 4: Connected to channel 1 Rx D-sub input via RS-232

JP2 Rx Jumper

- Pin 1: Connected to channel 1 Rx D-sub input via RS-232
- Pin 2: Connected to channel 1 Rx to Z180
- Pin 3: Connected to Tx output on ATmega328P
- Pin 4: Connected to channel 1 Tx D-sub output via RS-232

JP1 & JP2 function

- Pin 1 to Pin 2: Tx and Rx from Z180 connected to RS-232 D-sub
- Pin 2 to Pin 3: Tx and Rx from Z180 connected to Rx and Tx on ATmega328P
- Pin 3 to Pin 4: RS-232 D-sub connected to Rx and Tx on ATmega328P

JP3

- Connection from Z180 CKS to SCK on ATmega328P SPI interface

JP4

- Connection from Z180 RXS to MISO on ATmega328P SPI interface

JP5

- Connection from Z180 TXS to MOSI on ATmega328P SPI interface

JP6

- Connection from bit 2 on output port 0x44 to SS on ATmega328P SPI interface

PJ3, JP4, JP5, JP6 function

- All jumpers in place to connect Z180 SPI interface to ATmega328P SPI interface

JP7 EPROM/FLASH jumper

- Pin 1 connected to +5V
- Pin 2 connected to pin 31 on U1
- Pin 3 connected to /WR on Z180

JP7 Function

- Pin 1 to Pin 2: select EPROM in U1
- Pin 2 to Pin 3: select FLASH in U1

JP8 ATmega328P reset function

- Pin 1: connected to RS-232 D-sub channel 1 DSR via RS-232 to logic and a 100nF capacitor
- Pin 2: connected to ATmega328P /RESET
- Pin 3: connection from bit 3 on output port 0x44 to control reset from Z180

JP8 Function

- Pin 1 to Pin 2: reset of ATmega328P from DSR on serial channel 1
- Pin 2 to Pin 3: reset of ATmega328P from Z180 by setting bit 3 to "1" on output port 0x44

Port functions

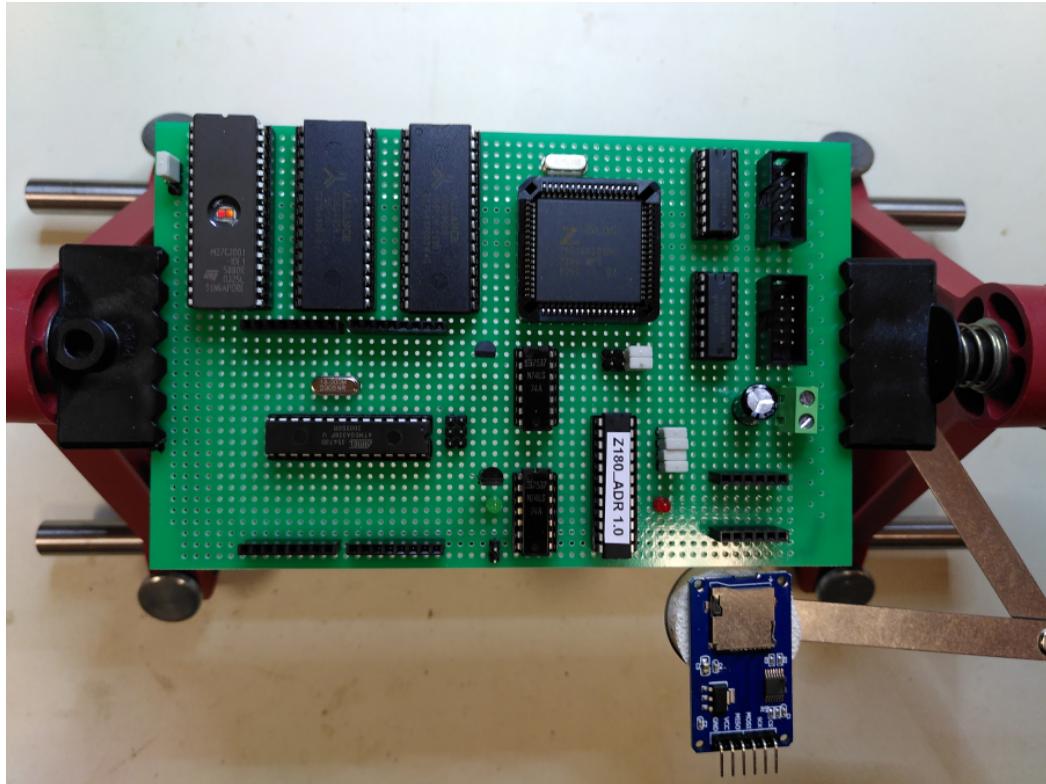
The function is selected by writing to the specified port (data bits, don't care)

- Write to port 0x40: select EPROM/FLASH in lower 256KB physical memory (data bits, don't care), set at reset
- Write to port 0x41: select RAM in lower 256KB physical memory (data bits, don't care)
- Write to port 0x42: LED off (data bits, don't care), set at reset
- Write to port 0x43: LED on (data bits, don't care)
- Write to port 0x44 using data bit 0 - bit 3, all are set to 0 at reset
 - - data bit 0 set to "1": select SD0 SPI FLASH
 - - data bit 1 set to "1": select SD1 SPI FLASH
 - - data bit 2 set to "1": select ATmega328P SPI interface
 - - data bit 3 set to "1": reset from Z180 to ATmega328P (if JP8 pin 2 and pin 3 are connected)

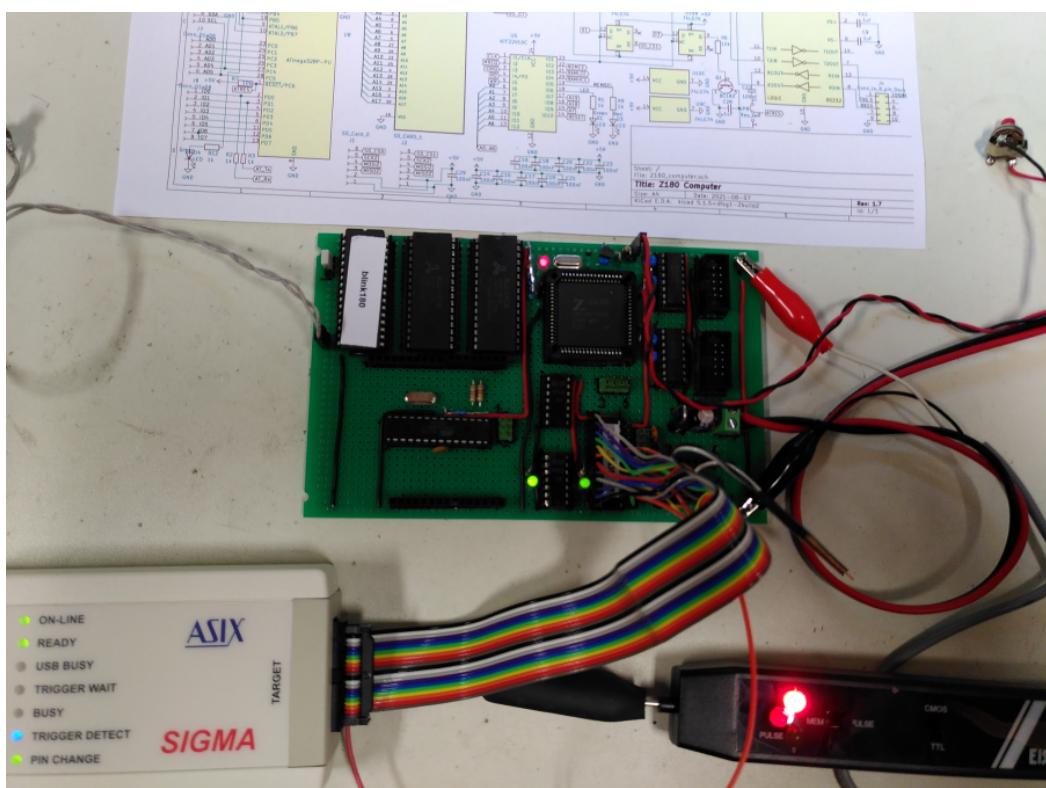
Board layout

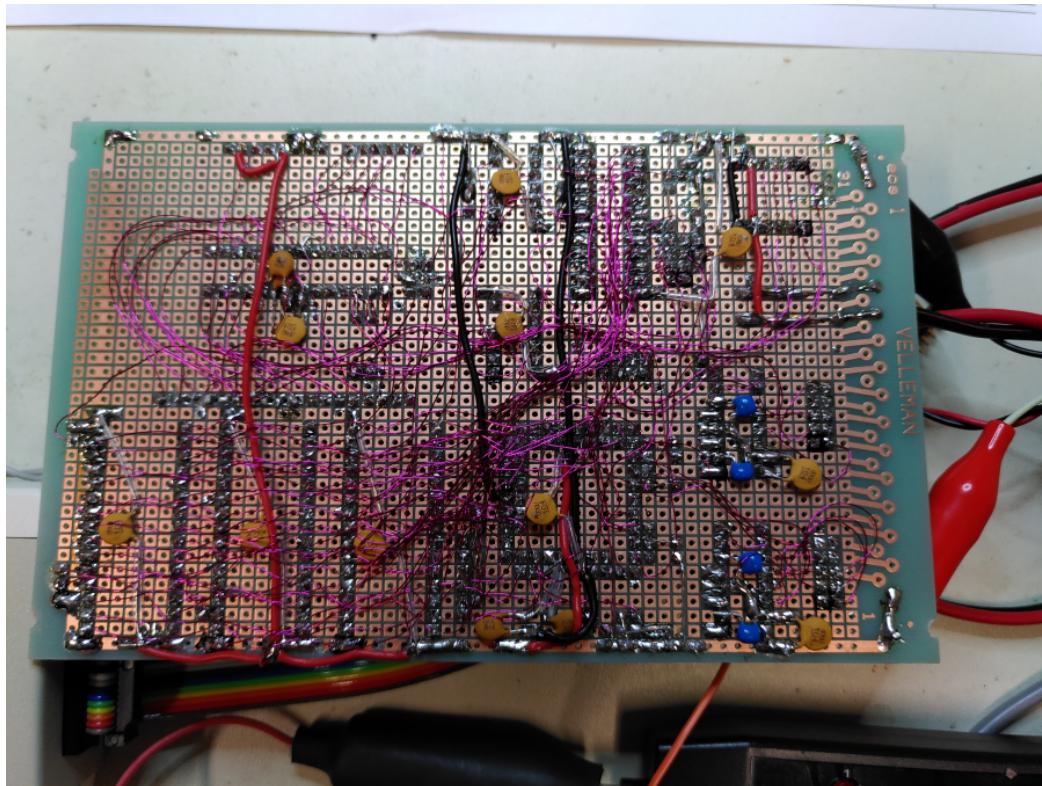
2021-07-15, updated 2021-07-20

Z180 computer



2021-08-09





KiCAD

2021-07-04

SnapEDA has a comprehensive symbol library: [Explore SnapEDA's Symbol & Footprint Libraries | SnapEDA \(https://www.snapeda.com/home/\)](https://www.snapeda.com/home/)

Added Z180 symbol library as described in: [Library management in KiCad version 5 - FAQ - KiCad.info Forums \(https://forum.kicad.info/t/library-management-in-kicad-version-5/14636\)](https://forum.kicad.info/t/library-management-in-kicad-version-5-faq-kicadinfo-forums/14636) from: [flypie/Z80-CPU-for-KiCAD: Z80 & Z180 CPU for KiCAD DIP/DIL/PLCC/QFP \(https://github.com/flypie/Z80-CPU-for-KiCAD\)](https://github.com/flypie/Z80-CPU-for-KiCAD)

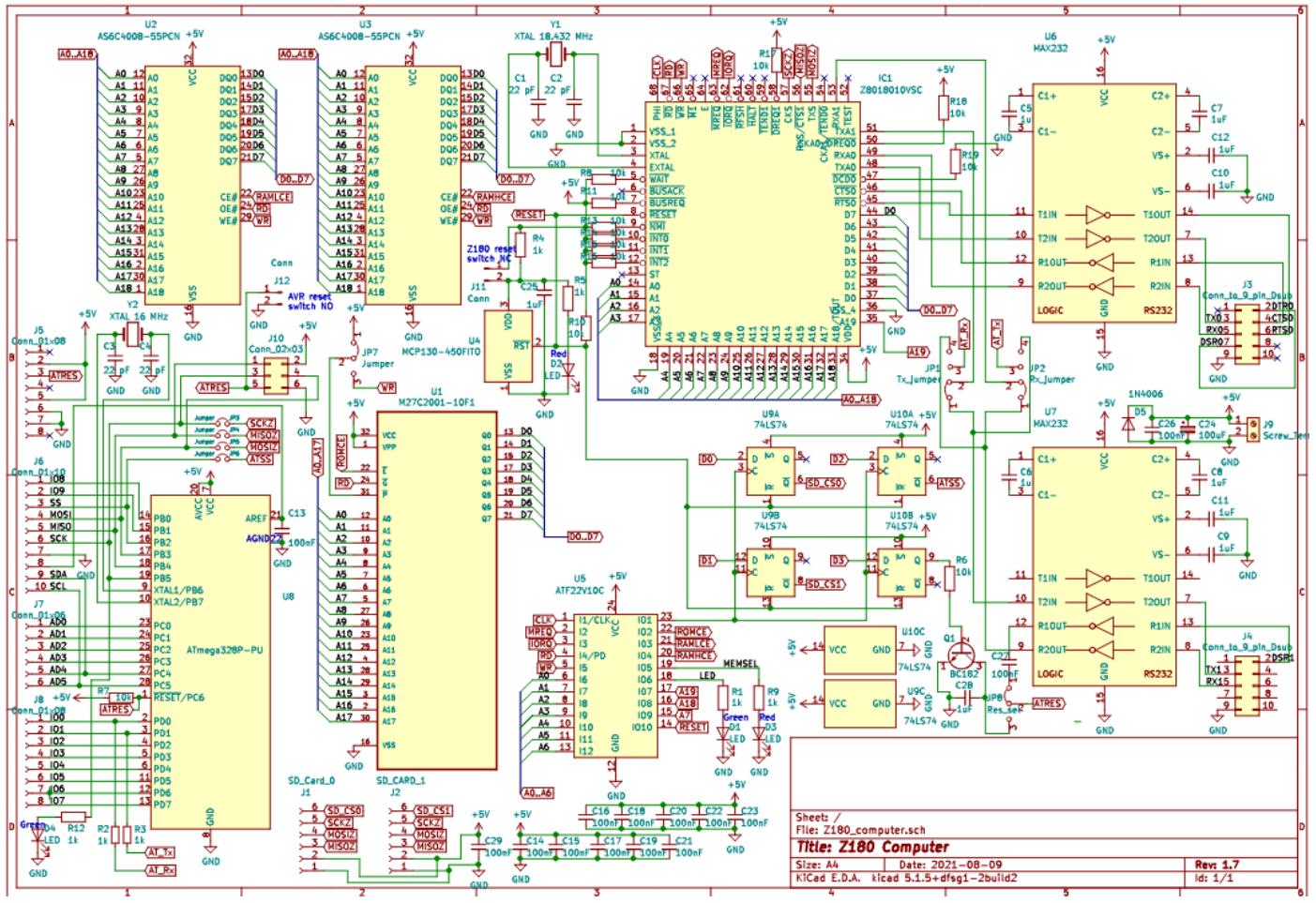
Added symbol for M27C2001 downloaded from: [M27C2001-10F1 footprint & symbol by STMicroelectronics | SnapEDA \(https://www.snapeda.com/part/M27C2001-10F1/STMicroelectronics/view-part/?welcome=home&ref=digikey&company=atHome\)](https://www.snapeda.com/part/M27C2001-10F1/STMicroelectronics/view-part/?welcome=home&ref=digikey&company=atHome)

(BS62LV1024 seems not to be available in any component library, just renamed 628128_DIP32 to BS62LV1024_DIP32 as it has the same pinout.)

Symbol for AS6C4008-55PCN is available in KiCAD.

Added symbol for ATF22V10 that was made for the Z80 computer.

Schematic, updated 2021-08-09:



File:Z180 computer schematic 2021-07-29.pdf

ATF22V10 PLD

2021-07-11, updated 2021-07-18, updated 2021-07-28 with reset and address A0 - A2

Address decoding is implemented with an ATF22V10 PLD.

Name	Z180_ADR;
Partno	U5;
Revision	1.1;
Date	2021-07-28;
Designer	hal;
Company	atHome;
Location	None;
Assembly	None;
Device	G22V10;

```
/*
 * Inputs: CPU clock, memory and io access signal and addresses
 */
Pin 1 = CLK;
Pin 2 = MREQ;
Pin 3 = IORQ;
Pin 4 = RD;
Pin 5 = WR;
Pin 6 = A0;
Pin 7 = A1;
Pin 8 = A2;
Pin 9 = A3;
Pin 10 = A4;
Pin 11 = A5;
Pin 13 = A6;
Pin 14 = RESET;
Pin 15 = A7;
Pin 16 = A18;
Pin 17 = A19;

/* Outputs: select register, i/o and memory select
 */
Pin 23 = CLKREG;
```

```

Pin 22 = ROMCE;
Pin 21 = RAMLCE;
Pin 20 = RAMHCE;
Pin 19 = MEMSEL;
Pin 18 = LED;

/*
 * Logic to control i/o addresses and memory
 */

/* Decode I/O addresses to select EPROM or RAM in lower memory */

ROMSEL = !IORQ & !WR & !A0 & !A1 & !A2 & !A3 & !A4 & !A5 & A6 & !A7; /* Write to 0x40 selects EPROM */
RAMSEL = !IORQ & !WR & A0 & !A1 & !A2 & !A3 & !A4 & !A5 & A6 & !A7; /* Write to 0x41 selects RAM */

MEMSEL.D = ((MEMSEL & !ROMSEL) # RAMSEL) & RESET;
MEMSEL.SP = 'b'0;
MEMSEL.AR = 'b'0;

/* Decode I/O addresses to control LED on/off */

LEDOFF = !IORQ & !WR & !A0 & A1 & !A2 & !A3 & !A4 & !A5 & A6 & !A7; /* Write to 0x42 turns LED off */
LEDON = !IORQ & !WR & A0 & A1 & !A2 & !A3 & !A4 & !A5 & A6 & !A7; /* Write to 0x43 turns LED on */

LED.D = ((LED & !LEDOFF) # LEDON) & RESET;
LED.SP = 'b'0;
LED.AR = 'b'0;

/* Latch data to 74LS74 based selection register */

CLKREG = CLK & !IORQ & !WR & !A0 & !A1 & A2 & !A3 & !A4 & !A5 & A6 & !A7; /* Write to 0x44 latches data */

/* Decode memory addresses */

/* High 512K RAM */
RAMHCE = !(!MREQ & A19);

/* Low 512K all RAM if no EPROM is selected (MEMSEL == 'b'1') */
/* RAM between 256K and 512K if EPROM is selected (MEMSEL == 'b'0') */
RAMLCE = !(!MREQ & (!(A19 & MEMSEL) # (A18 & !A19 & !MEMSEL)));

/* Low 256K EPROM if selected (MEMSEL == 'b'0') */
ROMCE = !(!MREQ & !A18 & !A19 & !MEMSEL);

```

And compiles with WinCUPL;

```

*****
Z180_ADR
*****

CUPL(WM)      5.0a Serial# 60008009
Device        g22v10 Library DLIB-h-40-1
Created       Wed Jul 28 11:57:49 2021
Name          Z180_ADR
Partno        U5
Revision      1.1
Date          2021-07-28
Designer      hal
Company       atHome
Assembly      None
Location      None

=====
Expanded Product Terms
=====

CLKREG =>
  !A0 & !A1 & A2 & !A3 & !A4 & !A5 & A6 & !A7 & CLK & !IORQ & !WR

LED.d =>
  A0 & A1 & !A2 & !A3 & !A4 & !A5 & A6 & !A7 & !IORQ & RESET & !WR
  # IORQ & LED & RESET
  # LED & RESET & WR
  # A0 & LED & RESET
  # !A1 & LED & RESET
  # A2 & LED & RESET
  # A3 & LED & RESET
  # A4 & LED & RESET
  # A5 & LED & RESET
  # A7 & LED & RESET
  # !A6 & LED & RESET

LED.ar =>
  0

LED.sp =>
  0

```

```

LEDOFF =>
  A0 & A1 & !A2 & !A3 & !A4 & !A5 & A6 & !A7 & !IORQ & !WR

LEDON =>
  A0 & A1 & !A2 & !A3 & !A4 & !A5 & A6 & !A7 & !IORQ & !WR

MEMSEL.d =>
  A0 & !A1 & !A2 & !A3 & !A4 & !A5 & A6 & !A7 & !IORQ & RESET & !WR
  # IORQ & MEMSEL & RESET
  # MEMSEL & RESET & WR
  # A0 & MEMSEL & RESET
  # A1 & MEMSEL & RESET
  # A2 & MEMSEL & RESET
  # A3 & MEMSEL & RESET
  # A4 & MEMSEL & RESET
  # A5 & MEMSEL & RESET
  # A7 & MEMSEL & RESET
  # !A6 & MEMSEL & RESET

MEMSEL.ar =>
  0

MEMSEL.sp =>
  0

RAMHCE =>
  A19 & !MREQ

RAMLCE =>
  A18 & !A19 & !MEMSEL & !MREQ
  # !A19 & MEMSEL & !MREQ

RAMSEL =>
  A0 & !A1 & !A2 & !A3 & !A4 & !A5 & A6 & !A7 & !IORQ & !WR

ROMCE =>
  !A18 & !A19 & !MEMSEL & !MREQ

ROMSEL =>
  !A0 & !A1 & !A2 & !A3 & !A4 & !A5 & A6 & !A7 & !IORQ & !WR

A7.oe =>
  0

A18.oe =>
  0

A19.oe =>
  0

CLKREG.oe =>
  1

LED.oe =>
  1

MEMSEL.oe =>
  1

RAMHCE.oe =>
  1

RAMLCE.oe =>
  1

RESET.oe =>
  0

ROMCE.oe =>
  1

```

?

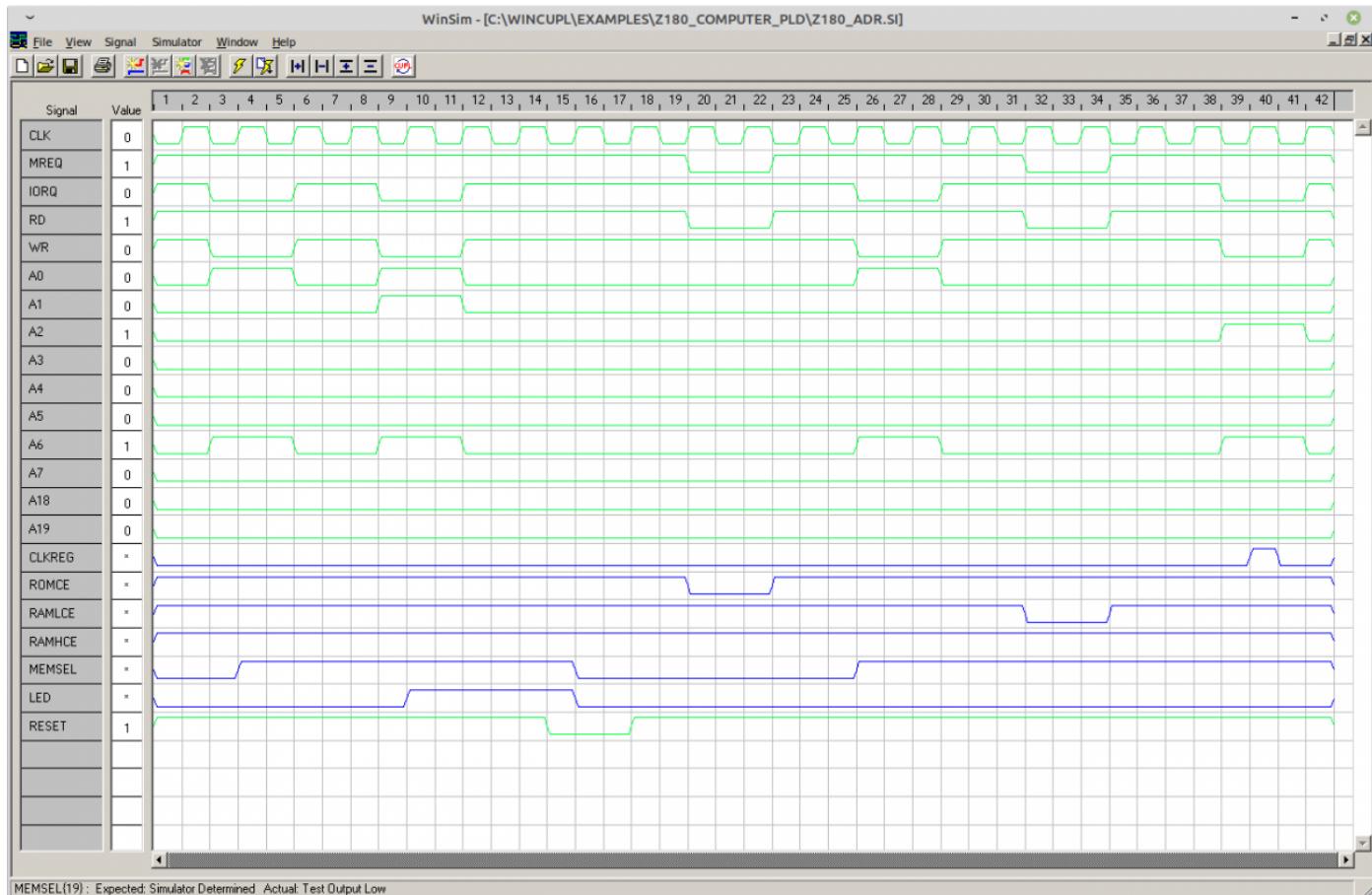
Symbol Table

Pin Pol	Variable Name	Ext	Pin	Type	Pterms Used	Max Pterms	Min Level
A0		---	6	V	-	-	-
A1		---	7	V	-	-	-
A2		---	8	V	-	-	-
A3		---	9	V	-	-	-
A4		---	10	V	-	-	-
A5		---	11	V	-	-	-
A6		---	13	V	-	-	-
A7		---	15	V	-	-	-
A18		---	16	V	-	-	-
A19		---	17	V	-	-	-
CLK		---	1	V	-	-	-
CLKREG		---	23	V	1	8	1
IORQ		---	3	V	-	-	-

LED		18	V	-	-	-
LED	d	18	X	11	16	1
LED	ar	18	X	1	1	1
LED	sp	18	X	1	1	1
LEDOFF	0	I	1	-	-	-
LEDON	0	I	1	-	-	-
MEMSEL		19	V	-	-	-
MEMSEL	d	19	X	11	16	1
MEMSEL	ar	19	X	1	1	1
MEMSEL	sp	19	X	1	1	1
MREQ	2	V	-	-	-	-
RAMHCE	20	V	1	14	1	
RAMLCE	21	V	2	12	1	
RAMSEL	0	I	1	-	-	-
RD	4	V	-	-	-	-
RESET	14	V	-	-	-	-
ROMCE	22	V	1	10	1	
ROMSEL	0	I	1	-	-	-
WR	5	V	-	-	-	-
A7	oe	15	D	1	1	0
A18	oe	16	D	1	1	0
A19	oe	17	D	1	1	0
CLKREG	oe	23	D	1	1	0
LED	oe	18	D	1	1	0
MEMSEL	oe	19	D	1	1	0
RAMHCE	oe	20	D	1	1	0
RAMLCE	oe	21	D	1	1	0
RESET	oe	14	D	1	1	0
ROMCE	oe	22	D	1	1	0

LEGEND D : default variable F : field G : group
 I : intermediate variable N : node M : extended node
 U : undefined V : variable X : extended variable
 T : function

The simulation output from winsim:



2021-07-19

Program and verify the ATF22V10C:

```
hal@LinuxServer:~/Z180_computer$ afterburner i -t ATF22V10C
PES info: Atmel ATF22V10C VPP=10.0 Timing: prog=10 erase=25
hal@LinuxServer:~/Z180_computer$ afterburner w -f Z180_ADR.jed -t ATF22V10C
```

```
| hal@LinuxServer:~/Z180_computer$ afterburner v -f Z180_ADR.jed -t ATF22V10C
```

Sometimes the first command has to be given a couple of times before the information shows up.

2021-07-26

The manual reset button does not reset the ATF22V10C, this has to be corrected as if reset is pressed while low RAM is selected, the EPROM start routines does not work. The registers in the Atmel ATF22V10Cs are designed to reset during power-up.

Design ideas, thoughts and questions

Z180 MMU

2021-08-12

The MMU itself is very simplistic. It divides the normal 16-bit address space into 3 sections: a section (called Common 0) that's always mapped to the beginning of physical memory; a section (called the Banked Area) which can be mapped anywhere in the 1Mb of physical memory; and a final section (called the Common 1 area) which can also be mapped anywhere in the 1Mb of physical memory.

The two banked areas can be made to start at any 4Kb region in the logical 64Kb memory space; though the Common 1 area should be made to start after the Banked Area.

The MMU is controlled via 3 x 8-bit registers in I/O space, at addresses at 0x38 to 0x3A.

Address 0x3A: CBAR specifies boundaries within the Z80180 64-KB logical address space for up to three areas: Common Area, Bank Area and Common Area 1.

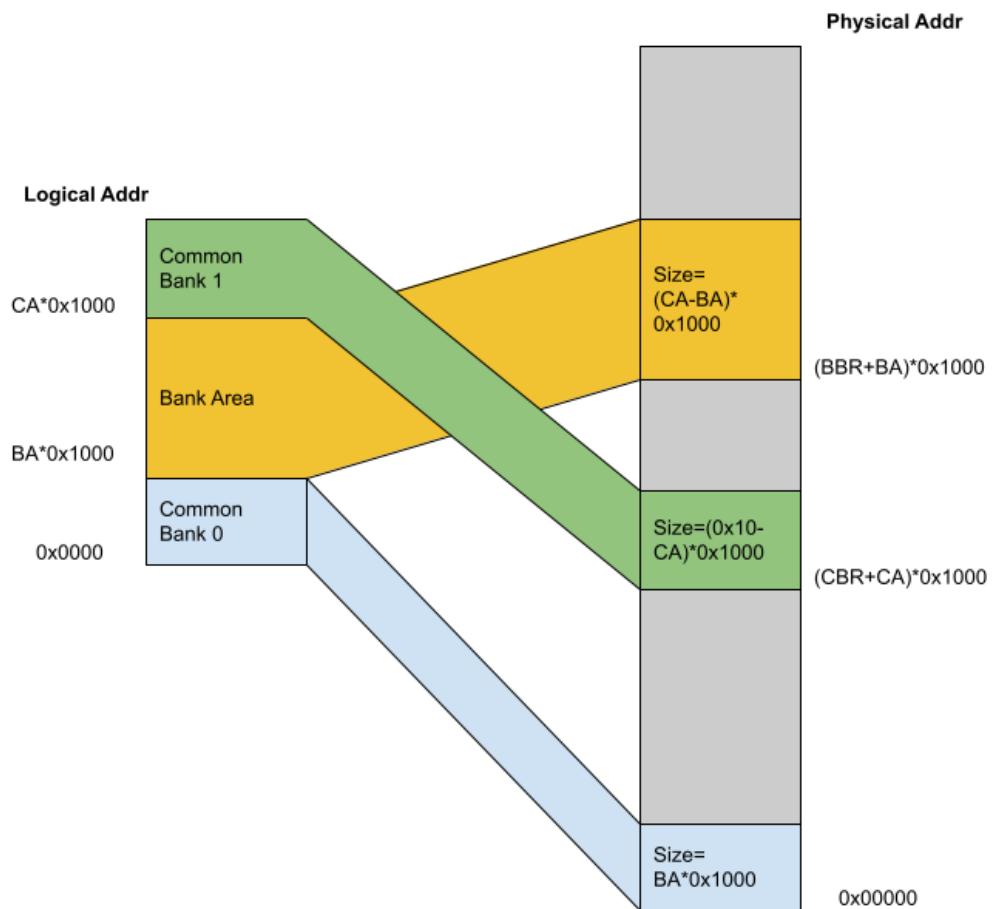
CA3-CA0:CA (bits 7-4)— CA specifies the start (Low) address (on 4 KB boundaries) for the Common Area 1, and also determines the most recent address of the Bank Area. All bits of CA are set to 1 during RESET.

BA-BA0 (bits 3-0)— BA specifies the start (Low) address (on 4-KB boundaries) for the Bank Area, and also determines the most recent address of the Common Area 0. All bits of BA are set to 1 during RESET.

Address 0x38: MMU Common Base Register (CBR)— CBR specifies the base address (on 4-KB boundaries) used to generate a 20-bit physical address for Common Area 1 accesses. All bits of CBR are reset to 0 during RESET.

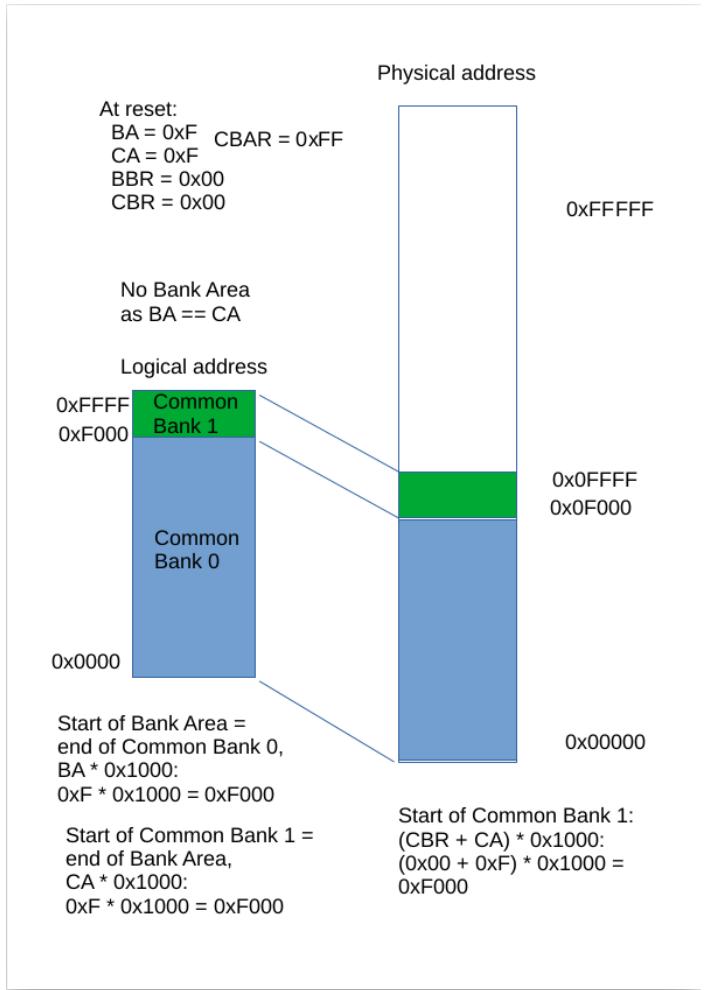
Address 0x39: BBR specifies the base address (on 4-KB boundaries) used to generate a 19-bit physical address for Bank Area accesses. All bits of BBR are reset to 0 during RESET .

Z180 MMU



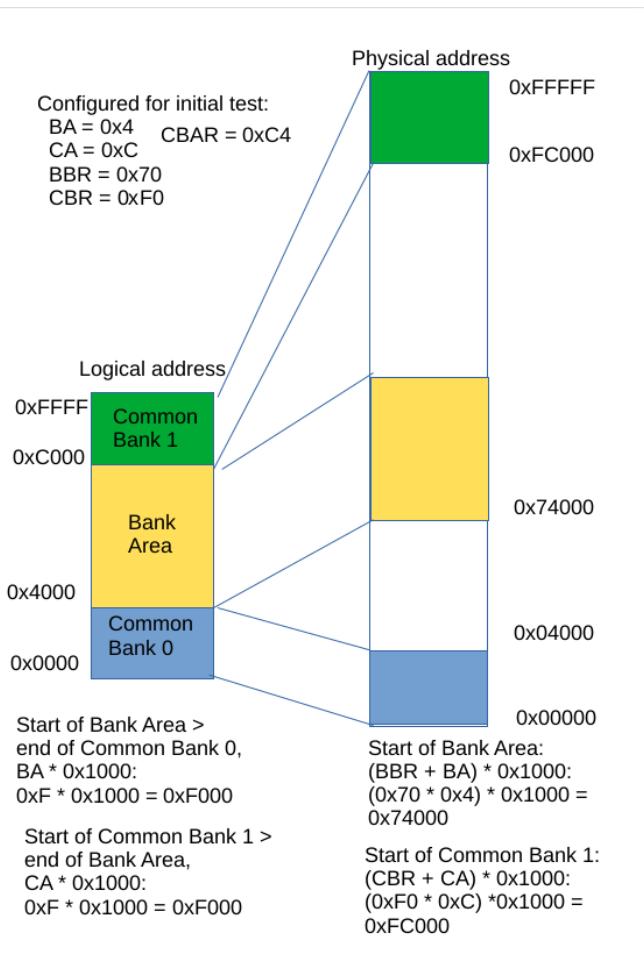
CA and BA are both reset to 0xf, which means that only the top 4Kb is mapped on boot-up, and since BBR and CBR are reset to 0, then this means that the top 4Kb is mapped to 0x0f000, which is good because the MMU can't be switched off.

Z180 MMU reset configuration:



CA and BA have the same value at reset and thus the size of the Bank Area is 0.

The test program will initially configure the MMU like this:



The physical memory layout of the Z180 computer:

- 0x00000 - 0x3FFFF: EPROM (or FLASH), 256KB, switchable between EPROM and RAM
- 0x00000 - 0x7FFFF: first RAM, 512KB, lower 256KB switchable between EPROM and RAM
- 0x80000 - 0xFFFF: second RAM, 512KB

References:

- [Z180 Memory Management](http://www.ganssle.com/articles/ammu.htm) (<http://www.ganssle.com/articles/ammu.htm>)
- [One Week Wonder: Z180 MMU-tiny](http://oneweekwonder.blogspot.com/2017/12/z180-mmu-tiny.html?m=1) (<http://oneweekwonder.blogspot.com/2017/12/z180-mmu-tiny.html?m=1>)

Clock, XTAL and Serial baudrate

2021-07-04

"The 10 MHz version is a good choice for starting out. If you use a 18.432 MHz crystal with that chip, it will run at half of that, or 9.216 MHz, and that will give serial port speeds of 9600 or 19.2k baud without any of the additional complications of external clocking." From: [The Z180 - oldcpusrus](https://sites.google.com/site/oldcpusrus/home/the-z180) (<https://sites.google.com/site/oldcpusrus/home/the-z180>).

2021-07-05

On-Chip Clock Generator. The Z8X180 contains a crystal oscillator and system clock generator. A crystal can be directly connected or an external clock input can be provided. In either case, the system clock is equal to one-half the input clock. For example, a crystal or external clock input of 8 MHz corresponds with a system clock rate of 4 MHz. From: [z8018x Family MPU User Manual - um0050.pdf](http://www.zilog.com/docs/z180/um0050.pdf) (<http://www.zilog.com/docs/z180/um0050.pdf>), page 168.

"Q: In our system, sometimes the PHI frequency is the same as XTAL frequency, not XTAL divided by two. Why? A: Please check the following points: Reset is held low at least 6 clock cycles. The status of ST line during reset. ST should not be tied low (ST line is the OUTPUT signal!" From: [00 Z180 Q & A - z180qa.pdf](http://www.zilog.com/docs/z180/appnotes/z180q.pdf) ([https://www.zilog.com/docs/z180/appnotes/z180q.pdf](http://www.zilog.com/docs/z180/appnotes/z180q.pdf))

a.pdf)

2021-08-11

Baud rate table: Z180 ASCI Baud Rate Options.pdf (<https://raw.githubusercontent.com/wwarthen/RomWBW/master/Doc/Z180%20ASCI%20Baud%20Rate%20Options.pdf>)

Driver: z88dk/libsrc/_DEVELOPMENT/target/scz180/device/asci at master · z88dk/z88dk (https://github.com/z88dk/z88dk/tree/master/libsrc/_DEVELOPMENT/target/scz180/device/asci)

2021-08-12

From: Z180 Q & A - z180faq.pdf (<https://www-users.cs.york.ac.uk/~pcc/Circuits/64180/docs/z180faq.pdf>)

Q: How to calculate baud rate?
A: Use the following formula to calculate:

$$\text{Baud rate} = \frac{\text{System Clock Speed}}{(\text{Clock factor}) \times (\text{PS bit}) \times (\text{Divide ratio})}$$

Where:
Clock factor - 16 or 64
PS bit - 10 or 30
SS0-SS2 - 1, 2, 4, 8, 16, 32 or 64

Q: What is the maximum baud rate for ASCII channels?
A: When using the internal clock for baud rate generation, it is:

$$\text{BRGmax} = \text{PHI}/(1 \times 10 \times 16)$$

Where: 1 is divider value, 10 is prescaler value,
and 16 is sampling rate.
When using the external clock, clock frequency on the external clock input is limited up to PHI/40, so:

$$\text{BRGmax} = \text{PHI}/(40 \times 16)$$

Where: 16 is sampling rate

Programming table:

CPU Clock	PHI: 9.216 Mhz	Baudrate	ASCI	Divisor	=	(Clock factor)	(PS bit)	(Divide ratio)
		57600	160	16		10	1	
		28800	320	16		10	2	
		19200	480	16		30	1	
		14400	640	16		10	4	
		9600	960	16		30	2	
		7200	1280	16		10	8	
		4800	1920	16		30	4	
		... etc.						

CSI/O based SPI (or i2c) interface to SD card

2021-07-07

The idea is to use the Z180 CSI/O interface to connect to an SD adapter via SPI. Just for fun an ATmega328P (the original Arduino CPU) could be included on the board and connected with SPI.

- A schematic on how to use CSI/O for SPI and i2c interfaces: schematic_sc126-v1.0-z180-motherboard-for-rc2014_20190629103757.pdf (https://smallcomputercentral.files.wordpress.com/2019/06/schematic_sc126-v1.0-z180-motherboard-for-rc2014_20190629103757.pdf)
 - The SPI interface is using CSI/O
 - The i2c interface is a "bitbanger" interface (same signals are also used for DS1302 Timekeeping Chip).
- Adapter to Micro SD card: Köp MicroSD-läsare 5V till rätt pris @ Electrokit (<https://www.electrokit.com/produkt/microsd-lasare-3-3-5v/>)
 - In-Depth Tutorial to Interface Micro SD Card Module with Arduino (<https://lastminuteengineers.com/arduino-micro-sd-card-module-tutorial/>)

References:

- [feilipu/z88dk-libraries](https://github.com/feilipu/z88dk-libraries): A collection of z80 libraries integrated with z88dk (<https://github.com/feilipu/z88dk-libraries>)
- Need I2C and SPI sample code ? (<https://groups.google.com/g/retro-comp/c/m8jUuTY8u34>)
- [z88dk/libsrc/_DEVELOPMENT/target/scz180](https://github.com/z88dk/z88dk/tree/master/libsrc/_DEVELOPMENT/target/scz180) at master · [z88dk/z88dk](https://github.com/z88dk/z88dk) (https://github.com/z88dk/z88dk/tree/master/libsrc/_DEVELOPMENT/target/scz180)
- [z88dk/libsrc/_DEVELOPMENT/target/scz180/device/csio/z180](https://github.com/z88dk/z88dk/tree/master/libsrc/_DEVELOPMENT/target/scz180/device/csio/z180) at master · [z88dk/z88dk](https://github.com/z88dk/z88dk) (https://github.com/z88dk/z88dk/tree/master/libsrc/_DEVELOPMENT/target/scz180/device/csio/z180)
- SC126 - Z180 Motherboard | Small Computer Central (<https://smallcomputercentral.wordpress.com/sc126-z180-motherboard-rc2014/>)
- Iterations of a SD Module « RC2014 (<https://rc2014.co.uk/1392/iterations-of-a-sd-module/>)
- SD Card Module with Arduino: How to Read/Write Data - Arduino Project Hub (<https://create.arduino.cc/projecthub/electropeak/sd-card-module-with-arduino-how-to-read-write-data-37f390>)
- Köp MicroSD-läsare 5V till rätt pris @ Electrokit (<https://www.electrokit.com/produkt/microsd-lasare-3-3-5v/>)
- Build Your Own Arduino & Bootload an ATmega Microcontroller - part 1 - [ElectroSchematics.com](https://www.electroschematics.com/build-arduino-bootload-atmega-microcontroller-part-1/) (<https://www.electroschematics.com/build-arduino-bootload-atmega-microcontroller-part-1/>)
- AVR151: Setup And Use of The SPI - Atmel-2585-Setup-and-Use-of-the-SPI_ApplicationNote_AVR151.pdf (https://ww1.microchip.com/downloads/en/AppNotes/Atmel-2585-Setup-and-Use-of-the-SPI_ApplicationNote_AVR151.pdf)
- How to use the "MicroSD Card Adapter" with the Arduino Uno | Michael Schoeffler (<https://mschoeffler.com/2017/02/22/how-to-use-the-microsd-card-adapter-with-the-arduino-uno/>)
- Arduino - SDCardNotes (<https://www.arduino.cc/en/Reference/SDCardNotes>)
- SOLVED. Nrf24 (Mirf lib) + Micro SD-card works OK together - Using Arduino / Storage - Arduino Forum (<https://forum.arduino.cc/t/solved-nrf24-mirf-lib-micro-sd-card-works-ok-together/347787/9>)

2021-07-08

- 8 bit latch with reset and inverted output, to select SPI channel (and maybe also to drive status LEDs).
 - 8 bit not easily available, but 2x 74LS74 (which I have) gives 4 latched outputs.
 - [Dual D-Type Positive-Edge -Triggered Flip-Flops With Preset And Clear datasheet - sn54ls74a-sp.pdf](https://www.ti.com/lit/ds/symlink/sn54ls74a-sp.pdf) (https://www.ti.com/lit/ds/symlink/sn54ls74a-sp.pdf?ts=1625736091583&ref_url=https%253A%252F%252Fwww.google.com%252F)

AVR / Arduino controller

2021-07-14

See: [AVR Programming](#)

- [Pocket AVR Programmer Hookup Guide](https://learn.sparkfun.com/tutorials/pocket-avr-programmer-hookup-guide) - learn.sparkfun.com (https://learn.sparkfun.com/tutorials/pocket-avr-programmer-hookup-guide?_ga=2.127691909.94672799.1626256475-796128395.1619009331)
- [Installing an Arduino Bootloader](https://learn.sparkfun.com/tutorials/installing-an-arduino-bootloader) - learn.sparkfun.com (<https://learn.sparkfun.com/tutorials/installing-an-arduino-bootloader>)
- [Pocket AVR Programmer Hookup Guide](https://learn.sparkfun.com/tutorials/pocket-avr-programmer-hookup-guide/programming-via-arduino) - learn.sparkfun.com (<https://learn.sparkfun.com/tutorials/pocket-avr-programmer-hookup-guide/programming-via-arduino>)

The ATmega328P Tx and Rx pins may be connected to the Z180 ASCII 1 Rx and Tx pins, or directly to one of the external RS-232 connectors via one of the MAX232 converters.

- Schematic om DIY Arduino using ATmega328P: [Build Your Own Arduino & Bootload an ATmega Microcontroller - part 1 - ElectroSchematics.com](https://www.electroschematics.com/build-arduino-bootload-atmega-microcontroller-part-1/) (<https://www.electroschematics.com/build-arduino-bootload-atmega-microcontroller-part-1/>)
- Arduino - Setting up an Arduino on a breadboard (<https://www.arduino.cc/en/Main/Standalone>)

2021-07-15

Added a 2x3 pin header for the SPI interface.

2021-07-18

Made a separate reset of ATmega328P controlled by the Z180 through bit 3 of the 74LS74 latches. Also moved the LED from this pin to the ATF22V10 pin 10.

2021-07-28

Reset from serial channel added.

- [Arduino Playground - DisablingAutoResetOnSerialConnection](https://playground.arduino.cc/Main/DisablingAutoResetOnSerialConnection) (<https://playground.arduino.cc/Main/DisablingAutoResetOnSerialConnection/>)

2021-08-03

- [The Full Arduino Uno Pinout Guide \[including diagram\]](https://www.circuito.io/blog/arduino-uno-pinout/) (<https://www.circuito.io/blog/arduino-uno-pinout/>)

- upload.wikimedia.org/wikipedia/commons/c/c9/Pinout_of_ARDUINO_Board_and_ATMega328PU.svg (https://upload.wikimedia.org/wikipedia/commons/c/c9/Pinout_of_ARDUINO_Board_and_ATMega328PU.svg)

Programming guides:

- [Burning the ATmega328p Bootloader](https://www.circuito.io/blog/atmega328p-bootloader/) (<https://www.circuito.io/blog/atmega328p-bootloader/>)
- [Burn bootloader to atmega328 without pc - Using Arduino / Microcontrollers - Arduino Forum](https://forum.arduino.cc/t/burn-bootloader-to-atmega328-without-pc/262139/3) (<https://forum.arduino.cc/t/burn-bootloader-to-atmega328-without-pc/262139/3>)
- [Overview | Standalone AVR Chip Programmer | Adafruit Learning System](https://learn.adafruit.com/standalone-avr-chip-programmer) (<https://learn.adafruit.com/standalone-avr-chip-programmer>)
- [adafruit/Standalone-Arduino-AVR-ISP-programmer: A standalone programmer for mass-programming AVR chips](https://github.com/adafruit/Standalone-Arduino-AVR-ISP-programmer) (<https://github.com/adafruit/Standalone-Arduino-AVR-ISP-programmer>)
- [Atmega328P-PU Bootloader \(Optiboot\) Burning Guide : 12 Steps - Instructables](https://www.instructables.com/Atmega328P-PU-Bootloader-(Optiboot)-Burning-Guide-12-Steps/) ([https://www.instructables.com/Atmega328P-PU-Bootloader-\(Optiboot\)-Burning-Guide-12-Steps/](https://www.instructables.com/Atmega328P-PU-Bootloader-(Optiboot)-Burning-Guide-12-Steps/))

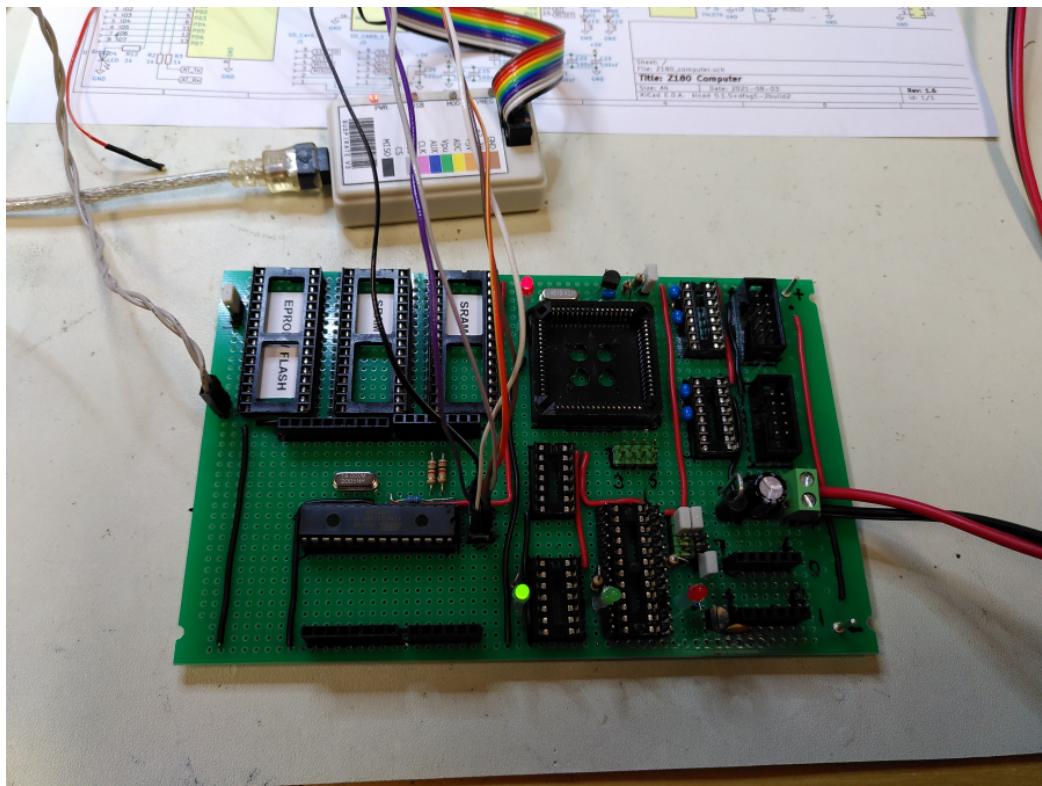
Added "built-in" LED on pin 19 (rearranged power LED)

- [Blink | Arduino](https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink) (<https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink>)

2021-08-04

Connected the ATmega328P to the 6 pin connector and uploaded the Blink sketch to the ATmega328P using Arduino IDE and the Bus Pirate as SPI interface.

AVR ISP programming



EPROM / FLASH jumper

2021-07-16

Added a EPROM / FLASH jumper as proposed by Bartosz Koziel on the Zilog Z-80 DIY group on Facebook.

Single step circuit (maybe TODO)

2021-08-05

An optional single step circuit would be good to have. In this stage a pin header with some signals from the CPU would be sufficient to add to the board.

[Single Step Instruction Circuit | Z80 Computer Project](https://z80project.wordpress.com/2014/02/16/single-step-instruction-circuit/) (<https://z80project.wordpress.com/2014/02/16/single-step-instruction-circuit/>)

Signals in pin header:

- M1
- WAIT
- more? TODO check

Test program

2021-08-05

The assembler z80asm is used: [AlbertVeli/z80asm: Clone of z80asm \(original is on Savannah\)](https://github.com/AlbertVeli/z80asm) (<https://github.com/AlbertVeli/z80asm>)

Makefile:

```
blink180.bin : blink180.z80
    date +'    db ", Built %F %R"' > built180.z80
    z80asm --list=blink180.lst --output=blink180.bin blink180.z80
```

blink180.lst, updated 2021-08-09

```
# File blink180.z80
0000 ; Test program that blinks the LED
0000 ; using, no RAM, no interrupt
0000 boot:
0000 mloop:
0000 21 ff ff ld hl,0xffff
0003 3e 01 ld a,1
0005 d3 43 out(0x43),a ; LED on
0007 onloop:
0007 2b dec hl
0008 7c ld a,h
0009 b5 or l
000a c2 07 00 jp nz,onloop
000d 21 ff ff ld hl,0xffff
0010 3e 00 ld a,0
0012 d3 42 out(0x42),a ; LED off
0014 offloop:
0014 2b dec hl
0015 7c ld a,h
0016 b5 or l
0017 c2 14 00 jp nz,offloop
001a
001a c3 00 00 jp mloop
001d
001d built:
001d include "built180.z80"
001d .. db ", Built 2021-08-08 17:25"
# End of file built180.z80
0035 00 db 0
0036
# End of file blink180.z80
0036
```

Program EPROM

```
L0001: Welcome to Elnec programmers control program PG4UW.
L0002: Version 3.15h/06.2015.
L0003:
L0004: Today is 05.08.2021, 14:06:08.
L0005:
L0006: Processor: Intel(R) Core(TM) i7-4790K CPU @ 4.00GHz Frequency: 3997\3997,77 MHz
L0007: CID: 0826-FFC6:08-00-27-75-6B-8A::57F928CB
L0008: Operating system: Windows 7 x64 (v6.1, Build 7601: Service Pack 1).
L0009: Physical RAM memory: 4096 MB or more.
L0010:
L0011:
L0012: >> 05.08.2021, 14:06:09
L0013: Default programmer: PREPROM-02aLV.
L0014: Scanning LPT1 (378h) port(s) for PREPROM-02aLV ..... found at port LPT1 (378h).
L0015: Establishing connection ... done.
L0016: Communication speed rate: reached 100% of maximum in this communication mode.
L0017:
L0018:
L0019:
L0020: >> 05.08.2021, 14:06:15
L0021: Selected device: STMicroelectronics M27C2001.
L0022:
L0023: Buffer operation "Checksum", time elapsed: 16 ms
L0024: Buffer checksum in range of [0h..3FFFFh]: 03FC0000h - Byte sum (x8), Straight
L0025:
L0026: >----- Begin of options list ----- (Program startup)
L0027:
L0028: |>----- Device operation options -----
L0029: | ----- Addresses -----
L0030: | Device start: "0000000000" h
L0031: | Device end: "000003FFFF" h
L0032: | Buffer start: "0000000000" h
L0033: | Split: "None"
L0034: | ----- Insertion test and/or ID check -----
L0035: | Insertion test: "Not supported"
L0036: | Device ID check error terminates the operation: "Enable"
L0037: | ----- Command execution -----
L0038: | Blank check before programming: "Disable"
L0039: | Verify after reading: "Enable"
L0040: | Verify: "Twice"
L0041: | Verify options: " 4.20V - 6.00V"
L0042: |<----- Device operation options -----
L0043:
L0044: <----- End of options list ----- (Program startup)
L0045:
L0046: Selected device: STMicroelectronics M27C2001.
L0047: Buffer checksum in range of [0h..3FFFFh]: 03FC0000h - Byte sum (x8), Straight
L0048:
L0049: >> 05.08.2021, 14:06:15
L0050: Buffer checksum type is set to "Byte sum (x8), Straight"
L0051: Buffer block(s) excluded from checksum calculation: Disabled
```

```

L0052:
L0053: >> Log file created at 05.08.2021 14:06:15
L0054: Log file name: C:\Users\hal\AppData\Roaming\Elnec\Pg4uw\report.rep
L0055: Log file mode: Append
L0056:
L0057: Buffer is erased with value FFh.
L0058:
L0059:
L0060: Buffer is erased with value FFh.
L0061: >> 05.08.2021, 14:08:31
L0062: Loading file (from "Load file" dialog window): C:\Users\hal\Desktop\EEPROM\blink180.bin
L0063: File format selection: automatic
L0064: File format: Binary
L0065: File loading successful. Bytes loaded 72 (00000048h).
L0066:
L0067: Buffer operation "Checksum", time elapsed: 15 ms
L0068: Buffer checksum in range of [0h..3FFFFh]: 03FBD1E6h - Byte sum (x8), Straight
L0069:
L0070: >> 05.08.2021, 14:08:55
L0071: Opened window "Program?" (parent window "PG4UW v3.15h/06.2015 - universal control...").
L0072:
L0073: >> 05.08.2021, 14:09:03
L0074: Closed window "Program?" (by pressing "Yes").
L0075:
L0076: >> 05.08.2021, 14:09:03
L0077: Programming device: STMicroelectronics M27C2001.
L0078: Buffer checksum in range of [0h..3FFFFh]: 03FBD1E6h - Byte sum (x8), Straight
L0079: Checking device ID ...
L0080: Programming device ...
L0081: Verifying device at VCCmax. ...
L0082: Verifying device at VCCmin. ...
L0083: Programming device - O.K.
L0084: Elapsed time: 0:02:21.0
L0085: Statistics info: Success:1 Failure:0 Other failure:0 Total:1

```

Components

2021-07-04

- 1x Z180 (68 pin PLCC) Zilog Z8018010VSC / Z180 MPU 10MHz [Z80180 Microprocessor Unit Product Specification - ps0140.pdf](https://www.zilog.com/docs/z180/ps0140.pdf) (<https://www.zilog.com/docs/z180/ps0140.pdf>)
- 1x M27C2001-10F1 256KB x 8(32 pin DIL) ST M27C2001 100nS M27C2001 Datasheet pdf - 2 MBIT (256KB X8) UV EPROM AND OTP ROM - SGS Thomson Microelectronics (http://www.datasheetsetcatalog.com/datasheets_pdf/M/2/7/C/M27C2001.shtml)
- (2x BS62LV1024PC-70 128KB x 8 (32 pin DIL) BSI BS62LV1024PC-70 70nS [BS62LV1024 datasheet](https://datasheetspdf.com/pdf-file/123572/BrillianceSemiconductor/BS62LV1024/1) (<https://datasheetspdf.com/pdf-file/123572/BrillianceSemiconductor/BS62LV1024/1>))
- 2x AS6C4008-55PCN 512KB x 8 (32 pin DIL) [AS6C4008-55PCN.pdf](http://www.farnell.com/datasheets/1911297.pdf) (<http://www.farnell.com/datasheets/1911297.pdf>) - ordered from eBay
- 2x MAX232 RS-232 Transciever [MAX232N Texas Instruments | Integrerade kretsar \(ICs\) | DigiKey](https://www.digikey.se/product-detail/sv/texas-instruments/MAX232N/296-1402-5-ND/277048) (<https://www.digikey.se/product-detail/sv/texas-instruments/MAX232N/296-1402-5-ND/277048>)
- 1x 22V10 PLDer (programmerbar logikenhet) IC 10 Makroceller 24-PDIP ATF22V10C-15PU [Microchip Technology | Integrerade kretsar \(ICs\) | DigiKey](https://www.digikey.se/product-detail/sv/microchip-technology/ATF22V10C-15PU/ATF22V10C-15PU-ND/1008580) (<https://www.digikey.se/product-detail/sv/microchip-technology/ATF22V10C-15PU/ATF22V10C-15PU-ND/1008580>)
- Köp MCP130-450FI TO-92 Övervakningskrets 5V till rätt pris @ Electrokit (<https://www.electrokit.com/produkt/mcp130-450fi-to-92-overvakningskrets-5v/>)
- Köp Kristall 18.432 MHz HC49/S till rätt pris @ Electrokit (<https://www.electrokit.com/produkt/kristall-18-432-mhz-hc49-s/>)
- ATmega48A, ATmega48PA, ATmega88A, ATmega88PA, ATmega168A, ATmega1688PA, ATmega328, ATmega328P datasheet - [ATmega48A_PA_88A_PA_168A_PA_328_P_DS_DS40002061B-1900559.pdf](https://www.mouser.se/datasheet/2/268/ATmega48A_PA_88A_PA_168A_PA_328_P_DS_DS40002061B-1900559.pdf) (https://www.mouser.se/datasheet/2/268/ATmega48A_PA_88A_PA_168A_PA_328_P_DS_DS40002061B-1900559.pdf)
- 2x 74LS74 (which I have) gives 4 latched outputs. [Dual D-Type Positive-Edge -Triggered Flip-Flops With Preset And Clear datasheet - sn54ls74a-sp.pdf](https://www.ti.com/lit/ds/symlink/sn54ls74a-sp.pdf) (https://www.ti.com/lit/ds/symlink/sn54ls74a-sp.pdf?ts=1625736091583&ref_url=https%253A%252Fwww.google.com%252F)

2021-08-09

Later a FLASH memory might be tested:

- [S93WD662/WD663 DS - 39SF020.pdf](http://www.alphacron.de/download/hardware/39SF020.pdf) (<http://www.alphacron.de/download/hardware/39SF020.pdf>)

Also ordered a 9.216 MHz crustal but as Z180 divides the XTAL frequency by 2 this is not needed:

- [Köp Kristall 9.216 MHz HC-49/S till rätt pris @ Electrokit](https://www.electrokit.com/produkt/kristall-9-216-mhz-hc-49-s/) (<https://www.electrokit.com/produkt/kristall-9-216-mhz-hc-49-s/>)

References

2021-07-03, updated 2021-08-12

- [z8018x Family MPU User Manual - um0050.pdf](http://www.zilog.com/docs/z180/um0050.pdf) (<http://www.zilog.com/docs/z180/um0050.pdf>)
- [Z80180 Microprocessor Unit Product Specification - ps0140.pdf](https://www.zilog.com/docs/z180/ps0140.pdf) (<https://www.zilog.com/docs/z180/ps0140.pdf>)
- [Z180 Q & A - z180qa.pdf](https://www.zilog.com/docs/z180/appnotes/z180qa.pdf) (<https://www.zilog.com/docs/z180/appnotes/z180qa.pdf>)
- [wwarthen/RomWBW: System Software for Z80/Z180 Computers](https://github.com/wwarthen/RomWBW) (<https://github.com/wwarthen/RomWBW>)
- [start \[RetroBrew Computers Wiki\]](https://www.retrobrewcomputers.org/doku.php) (<https://www.retrobrewcomputers.org/doku.php>)
- [boards:sbc:z180_mark_iv:z180_mark_iv \[RetroBrew Computers Wiki\]](https://www.retrobrewcomputers.org/doku.php?id=boards:sbc:z180_mark_iv:z180_mark_iv) (https://www.retrobrewcomputers.org/doku.php?id=boards:sbc:z180_mark_iv:z180_mark_iv)
- [The Z180 Interfaced with the SCC at 10 MHz - an0096.pdf](http://www.zilog.com/docs/appnotes/an0096.pdf) (<http://www.zilog.com/docs/appnotes/an0096.pdf>)
- [Z180_Technical_Manual_Jun88.pdf](http://www.bitsavers.org/components/zilog/z180/Z180_Technical_Manual_Jun88.pdf) (http://www.bitsavers.org/components/zilog/z180/Z180_Technical_Manual_Jun88.pdf)
- [Z180 Memory Management](http://www.ganssle.com/articles/ammu.htm) (<http://www.ganssle.com/articles/ammu.htm>)
- [One Week Wonder: Z180 MMu-tiny](http://oneweekwonder.blogspot.com/2017/12/z180-mm-tiny.html?m=1) (<http://oneweekwonder.blogspot.com/2017/12/z180-mm-tiny.html?m=1>)
- [mtdev79/z180emu: A portable full system emulator of Z180 based boards](https://github.com/mtdev79/z180emu) (<https://github.com/mtdev79/z180emu>)
- [cpnet-z80/Z180CSIO.md at master · durgadas311/cpnet-z80](https://github.com/durgadas311/cpnet-z80/blob/master/Z180CSIO.md) (<https://github.com/durgadas311/cpnet-z80/blob/master/Z180CSIO.md>)
- [Yet Another Z180 Project \(YAZ180\) | feilipu](https://feilipu.me/2016/05/23/another-z80-project/) (<https://feilipu.me/2016/05/23/another-z80-project/>)
- [feilipu/yaz180: YAZ180 - Modern Single Board Z180 Computer](https://github.com/feilipu/yaz180) (<https://github.com/feilipu/yaz180>)
- [Z180 CPU](http://www.z80.no/projects/Z180.html) ([https://www.z80.no/projects/Z180.html](http://www.z80.no/projects/Z180.html))
- [The Z180 - oldcpusrus](https://sites.google.com/site/oldcpusrus/home/the-z180) (<https://sites.google.com/site/oldcpusrus/home/the-z180>)
- [flypie/Z80-CPU-for-KiCAD: Z80 & Z180 CPU for KiCAD DIP/DIL/PLCC/QFP](https://github.com/flypie/Z80-CPU-for-KiCAD) (<https://github.com/flypie/Z80-CPU-for-KiCAD>)
- [Zilog Z180 Single Board Computer project - Page 1](https://www.eevblog.com/forum/microcontrollers/zilog-z180-single-board-computer-project/) (<https://www.eevblog.com/forum/microcontrollers/zilog-z180-single-board-computer-project/>)
- [Z-World BL1300.pdf](https://www.digi.com/resources/documentation/digidocs/pdfs/0190006_03.pdf) (https://www.digi.com/resources/documentation/digidocs/pdfs/0190006_03.pdf)
- [J2 : - phoenix_board.pdf](http://oric.free.fr/phoenix_board.pdf) (http://oric.free.fr/phoenix_board.pdf)

Retrieved from 'http://192.168.42.21/mediawiki/index.php?title=Z180_Computer&oldid=8353'

This page was last modified on 12 August 2021, at 17:49.