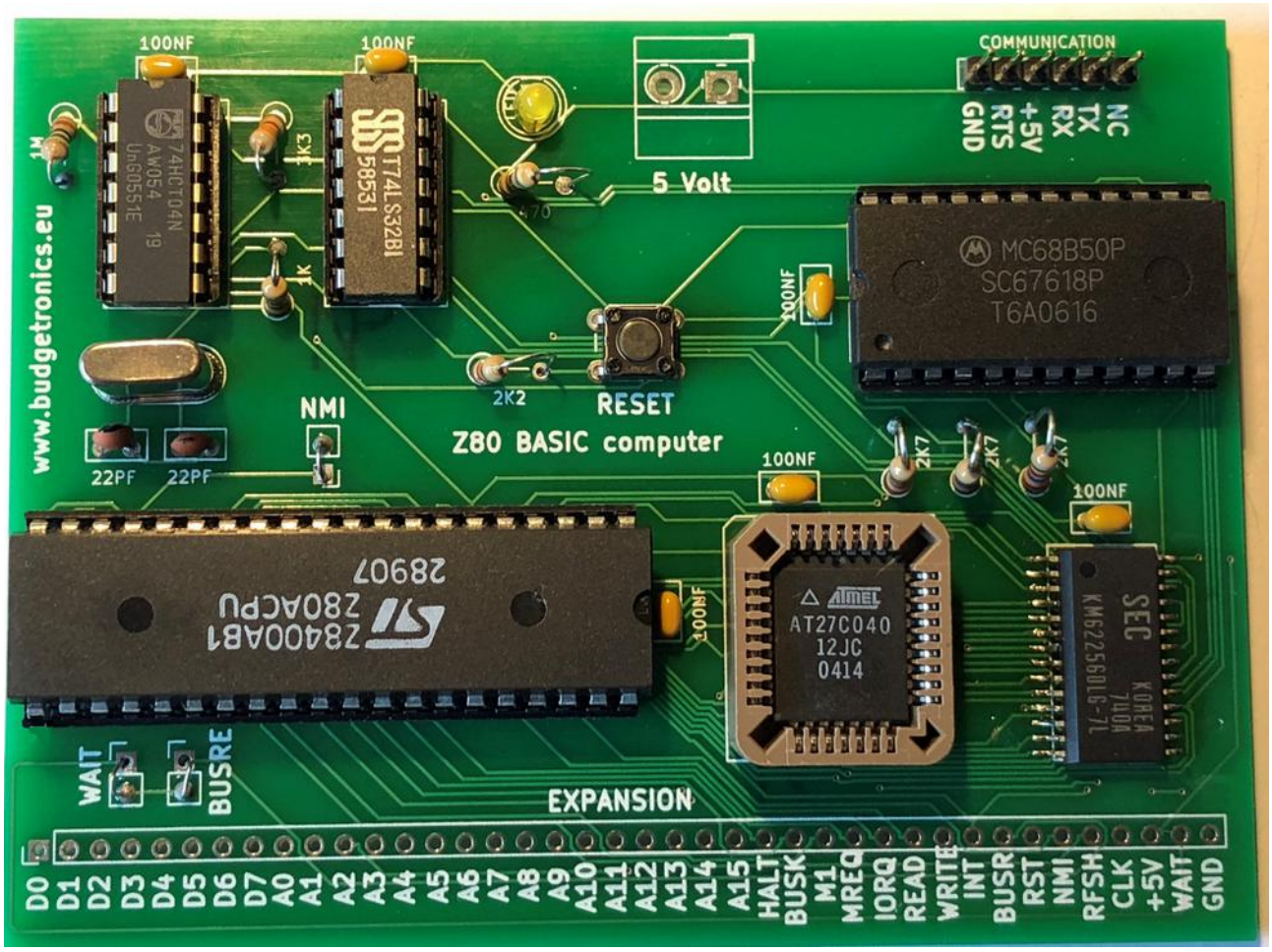
BudgeTronics

Z80 RETROCOMPUTER



Compact computer with Basic in ROM.

With original Z80 CPU!

Easy to build.

The Z80 retrocomputer building kit

Go back to the seventies and eighties of the last century and see for yourself how computing was in these days.

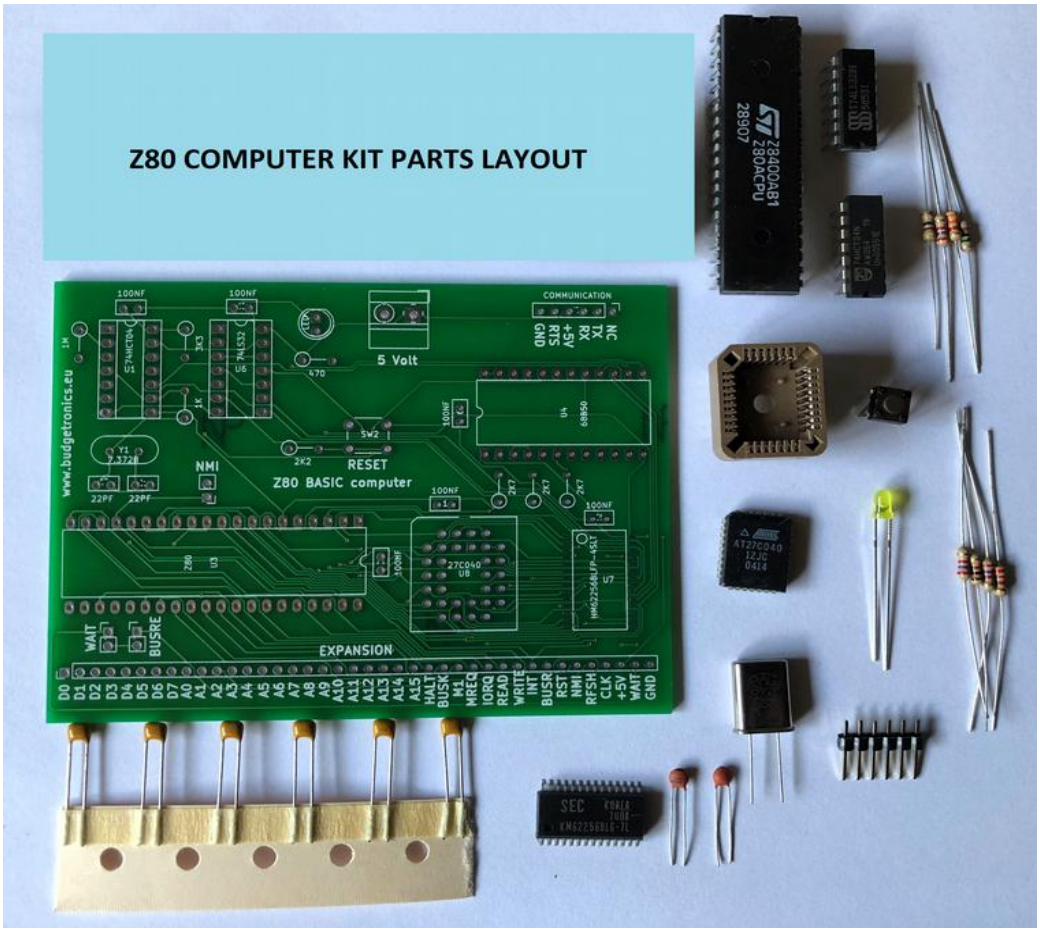
With this affordable building kit you can build your own Z80 retrocomputer with original Z80 CPU. The supplied ROM is already programmed with the BASIC language so it will be easy to program.

The computer has a RAM memory of 32K Byte and all signals from the CPU are available in the edge of the PCB. In this way you can add your own circuit designs if needed.

The output of the computer is serial and you need a TTL to USB serial cable and the free terminal program Tera-Term. See downloadsection for the software on the left side down on our website www.budgetronics..eu . You also can use your own terminal program or solution for displaying the computer output. In the download package you also find some Basic example programs to load into your computer using a terminal program.

To make this a stand alone computer you can add the ASCII video terminal building kit. In this way you add a VGA video output or composite video output and a PS/2 keyboard input. See menu Buidling kits in our website www.budgetronics.eu for more details about this kit.

CONTENTS BUILDING KIT



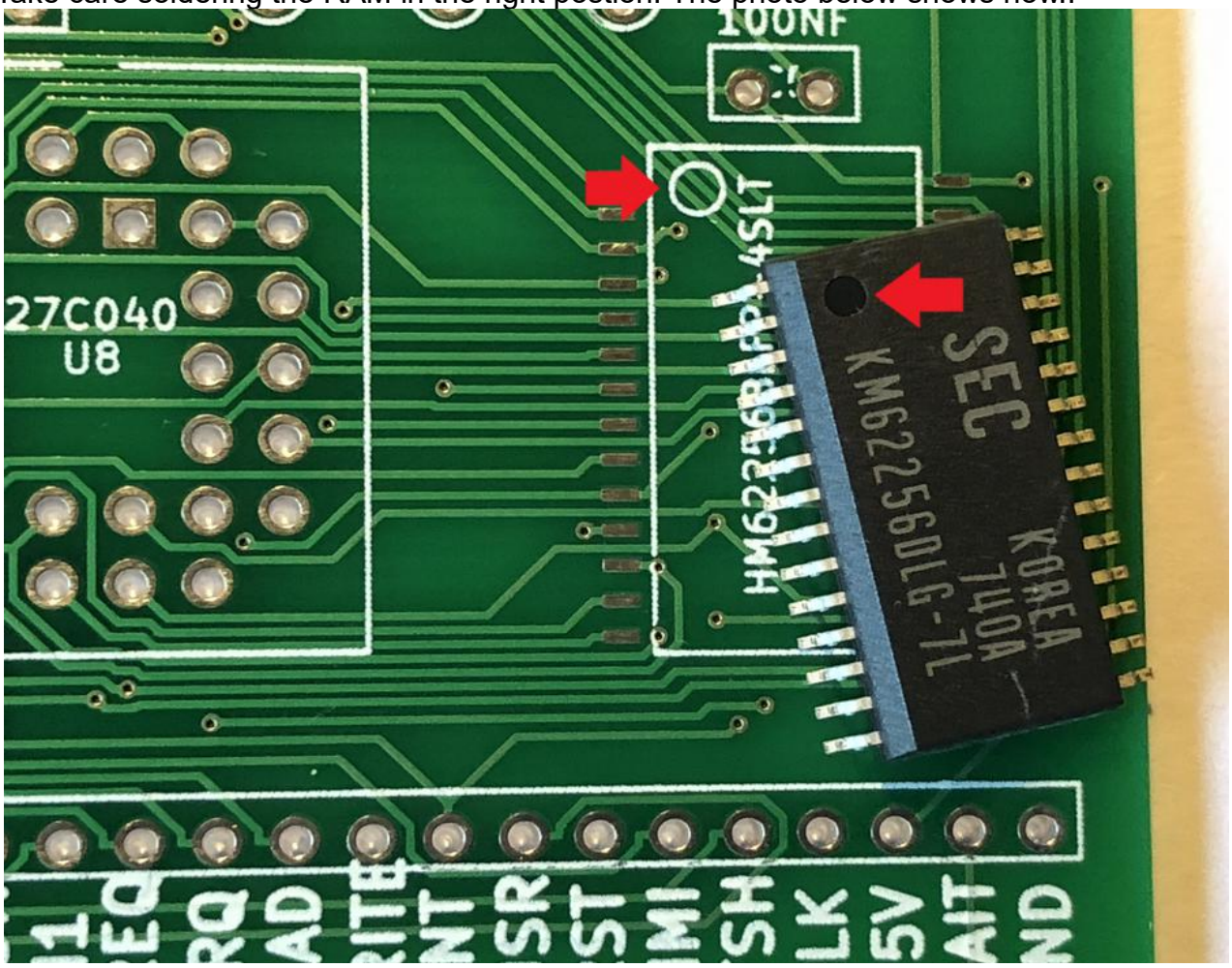
In the building kit you will find the following parts:

1x PCB
1x IC socket PLCC32
1x Original Z80 CPU
1x 74LS32 IC
1x 74HCT04 IC
1x SMD RAM IC KM62256 or similar
1x ROM AT27C040 with programmed Basic
1x 68B50 IC Asynchronous Communications Interface Adapter (ACIA) for serial communication.
3x resistor 2K7 (red, violet,red)
1x resistor 2k2 (red, red, red)
1x resistor 470 (yellow, violet, brown)
1x resistor 1K (brown, black, red)
1x resistor 3k3 (orange, orange, red)
1x resistor 1M (brown, black, green)
1x quartz 3.6864MHz
2x ceramic capacitor 22pF
6x ceramic capacitor 100nF (104)
1x 6 pins header
1x yellow 3mm LED
1x pushbutton for reset

Building the kit

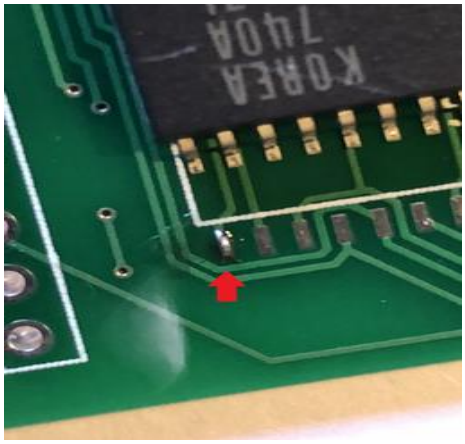
We begin soldering the SMD RAM KM62256. For those not familiar with soldering SMD parts it is not difficult if you follow the explanation here.

Take care soldering the RAM in the right postion. The photo below shows how..



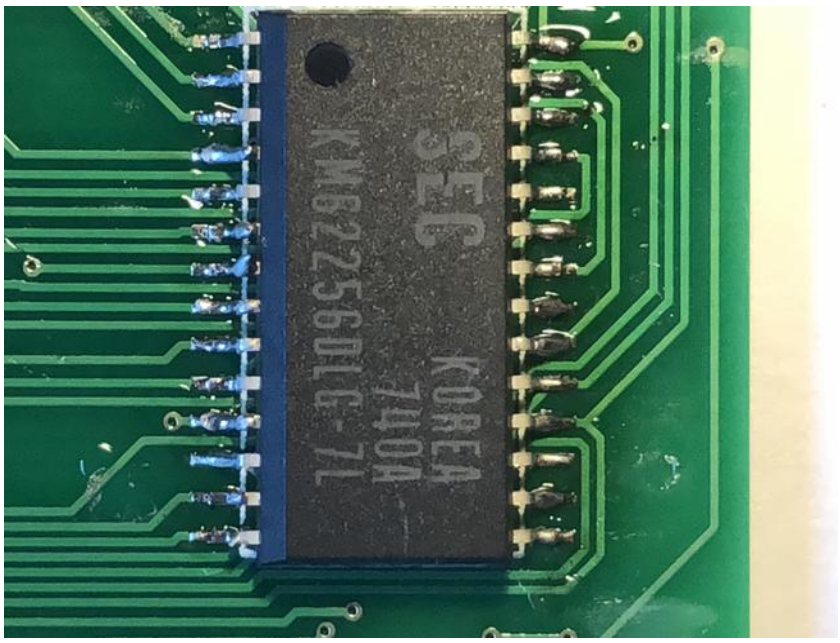
Circle on circle

Put a little solder on one of the corner connections like on the photo below.



Heat the solder and slide IC to the corner and heat IC pin and solder. Line the IC out with all the other connections. Solder the opposite corner only if all the pins are lined up correctly. After this it looks like the photo below.

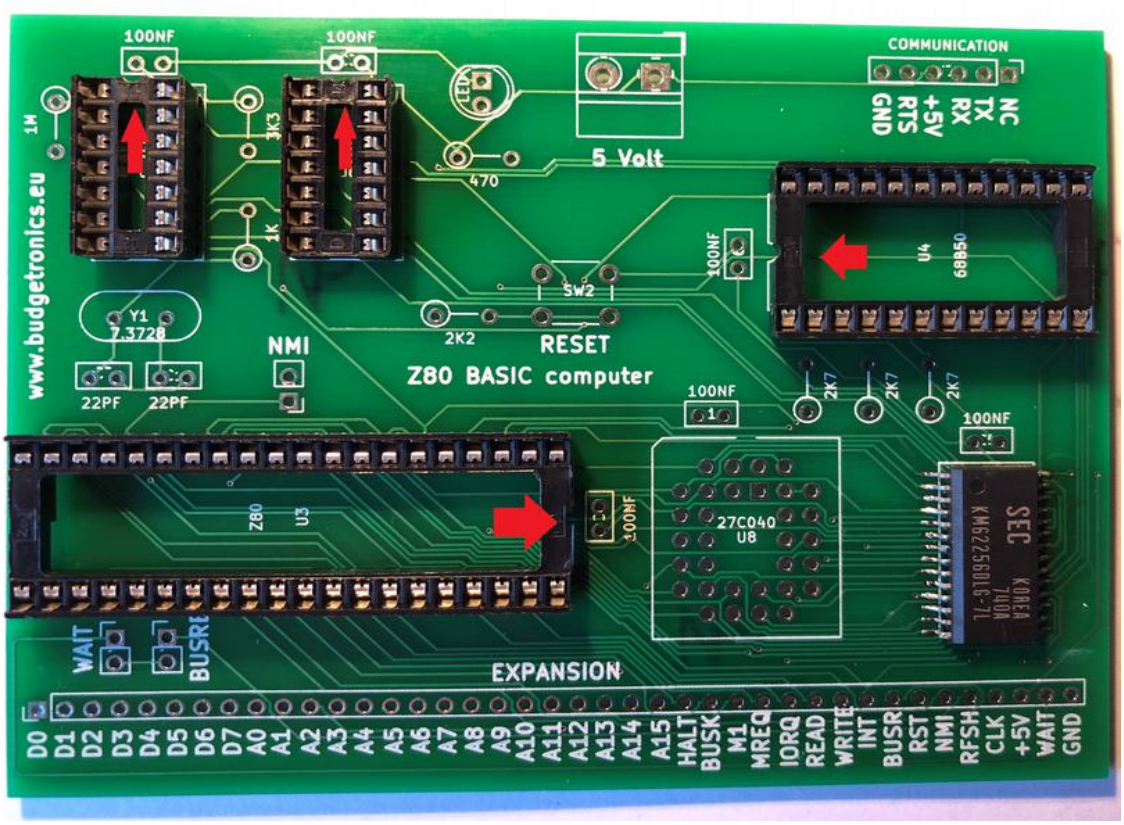
If all is lined up oke you solder all the other pins with a little bit of solder. Always look with a magnifying glass if al the connections are really made and if they do not make unwanted contact with other pins. After soldering the RAM IC it will look like the photo below (maybe even better if you are really good at soldering :-))



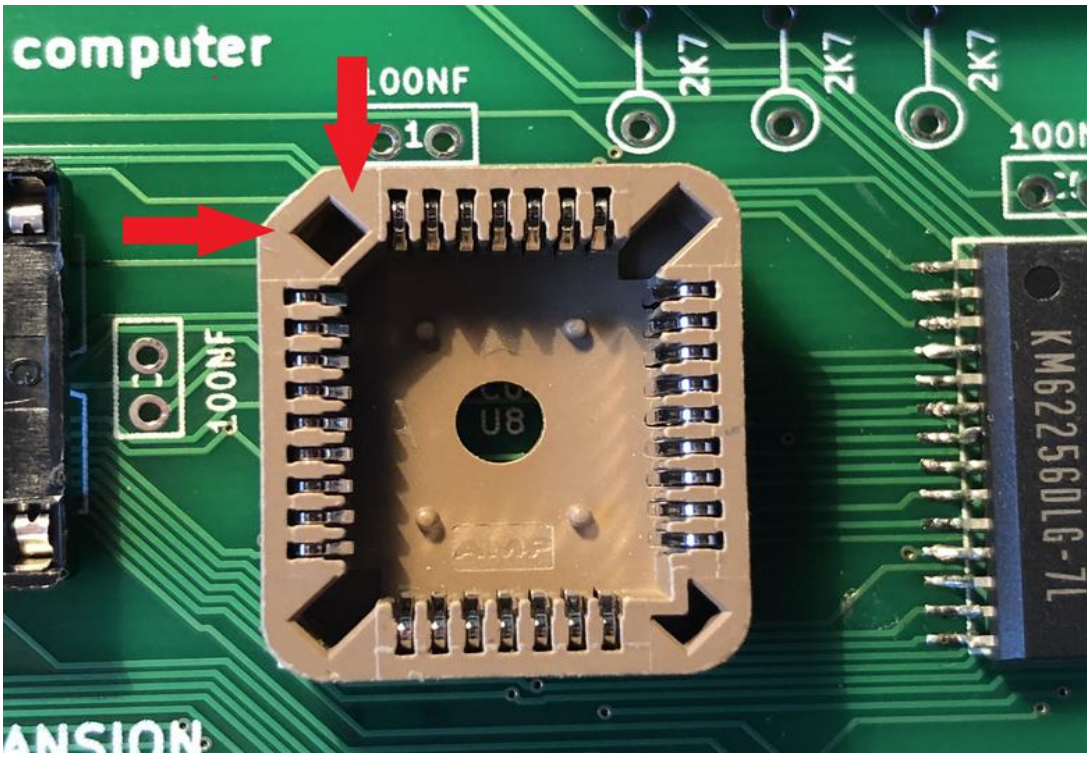
If this is done the remaining construction of the kit will be very easy. The most difficult part is done now.

If you want to use IC sockets (**NOT INCLUDED WITH THIS KIT only PLCC socket**) now is the time to solder the IC sockets and line them up with the PCB indicators. See red arrows. IC sockets are not really needed the IC's are very robust and can be soldered

without sockets.



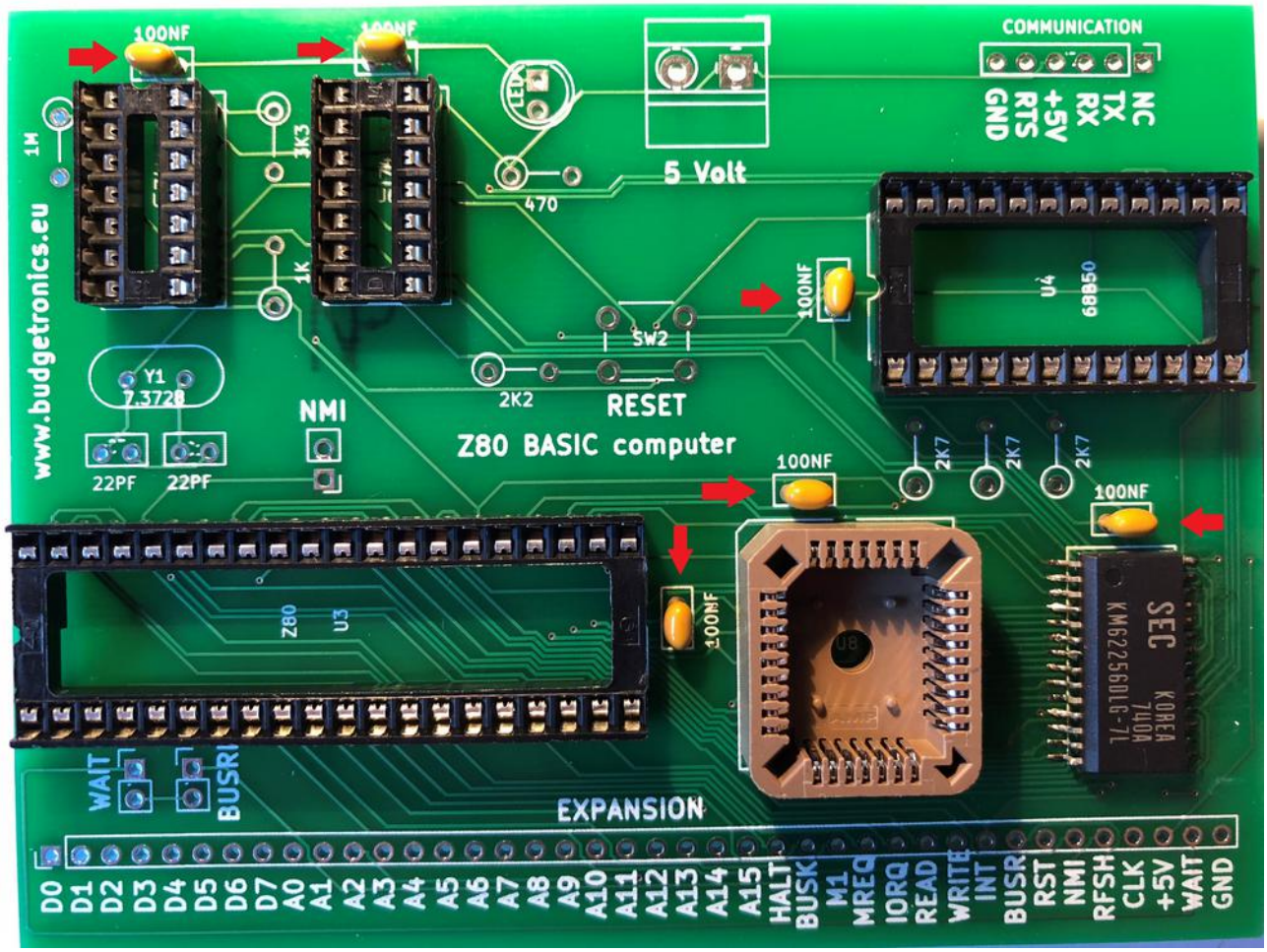
Solder the in the kit included PLCC32 socket and **make sure you do this the right way!** The PLCC socket can be soldered the wrong way and the ROM IC will not work in that case!! **Look very careful to the red arrow.**



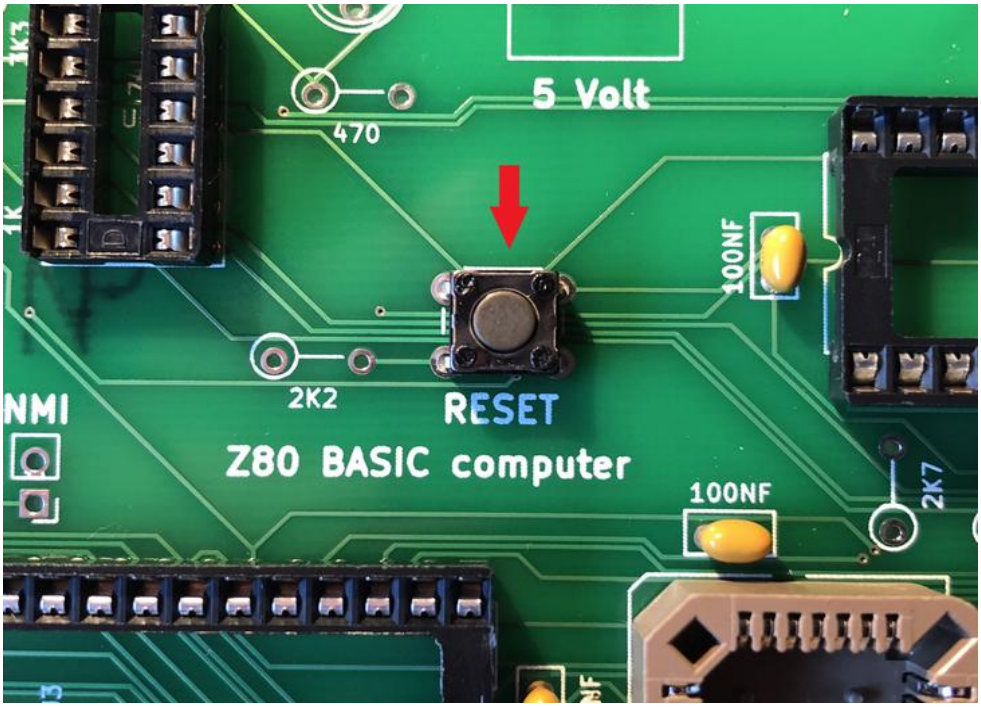
The small square come at the left side up as seen in the photo.

Connect the 6 x 100nf (code 104 or 100nf) capacitors as indicated with the red arrows

below.

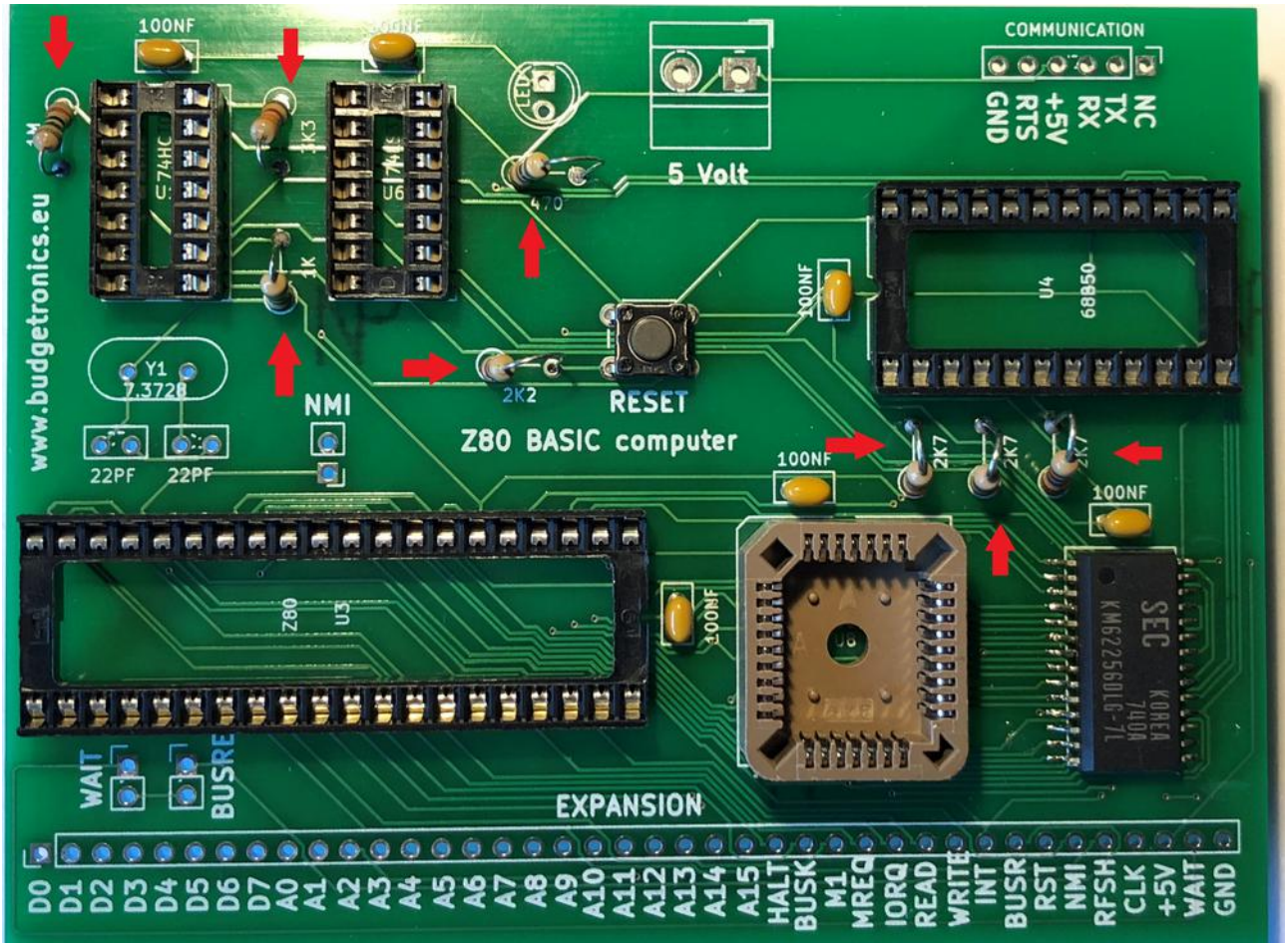


Connect the push button.

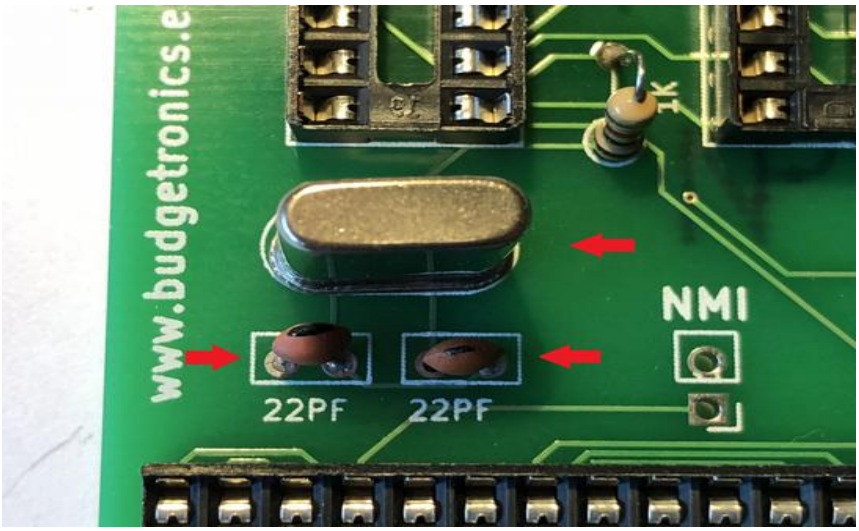


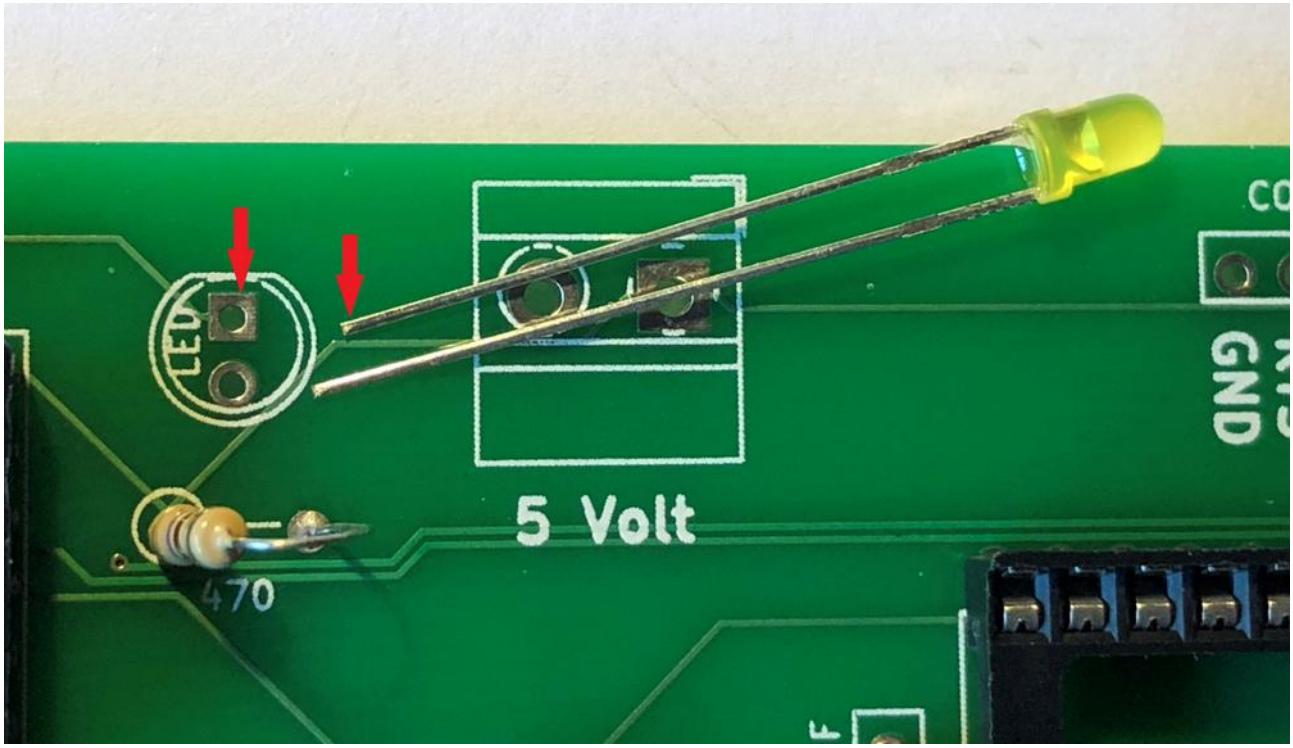
Solder the 8 resistors with the correct values in the right place. For the color codes see

description “Contents building kit” above.

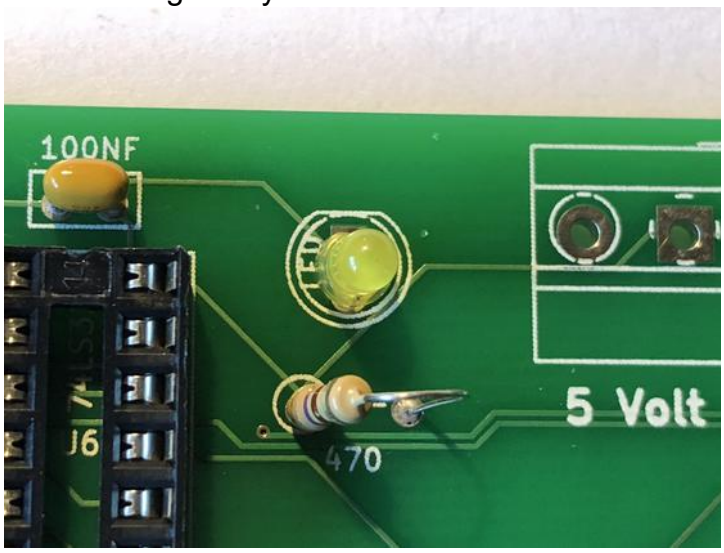


Connect the quartz of 3.6864MHz and the two 2x 22pf capacitors. On the PCB you will see 7,3728 but this is only for Z80 CPU's which can operate at 8 Mhz. If you want to try this in the future you can place a 7,3728 Mhz quartz and in that case you need to use 115200 Baud as a speed for the serial connection instead of 57600 Baud.

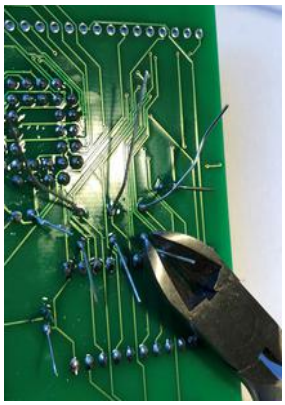




Connect the LED in the right way.

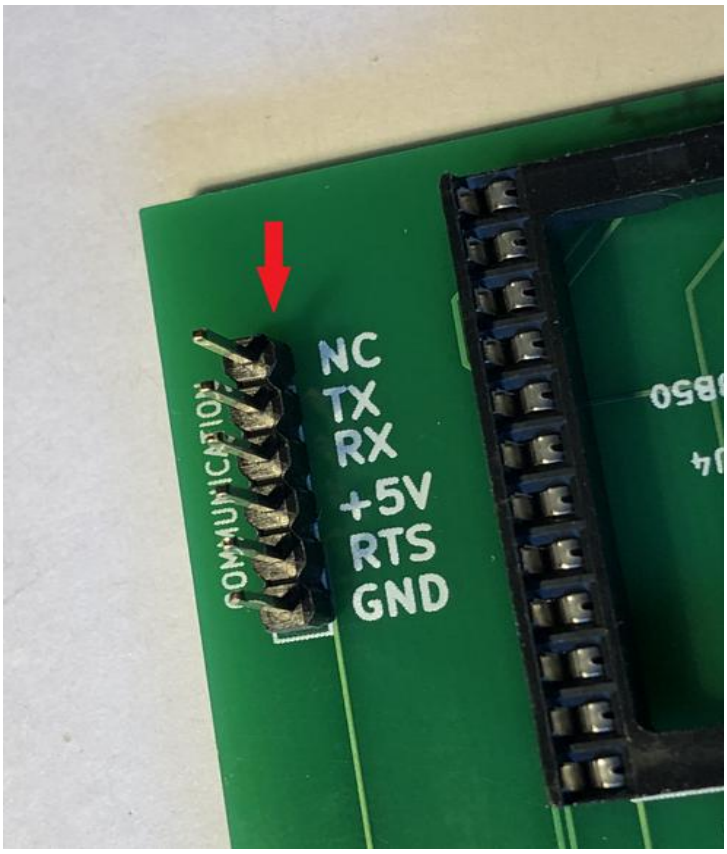


Do not forget to cut all the wires at the back of the PCB.

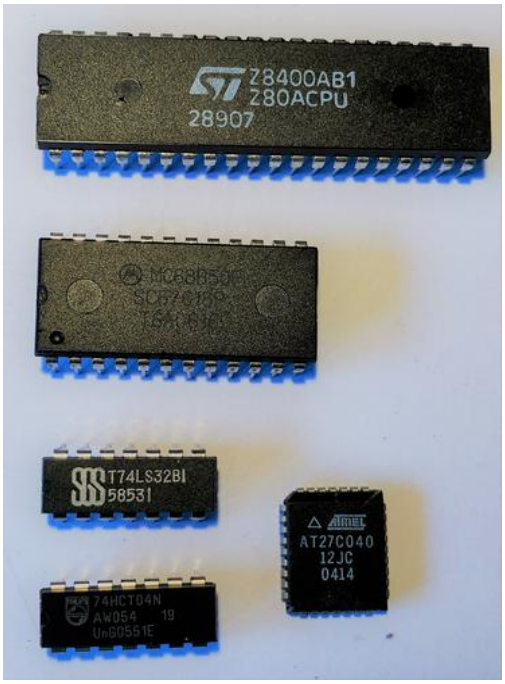


Keep some of the cut off wires because you will need some later.

Connect the 6 pins header.

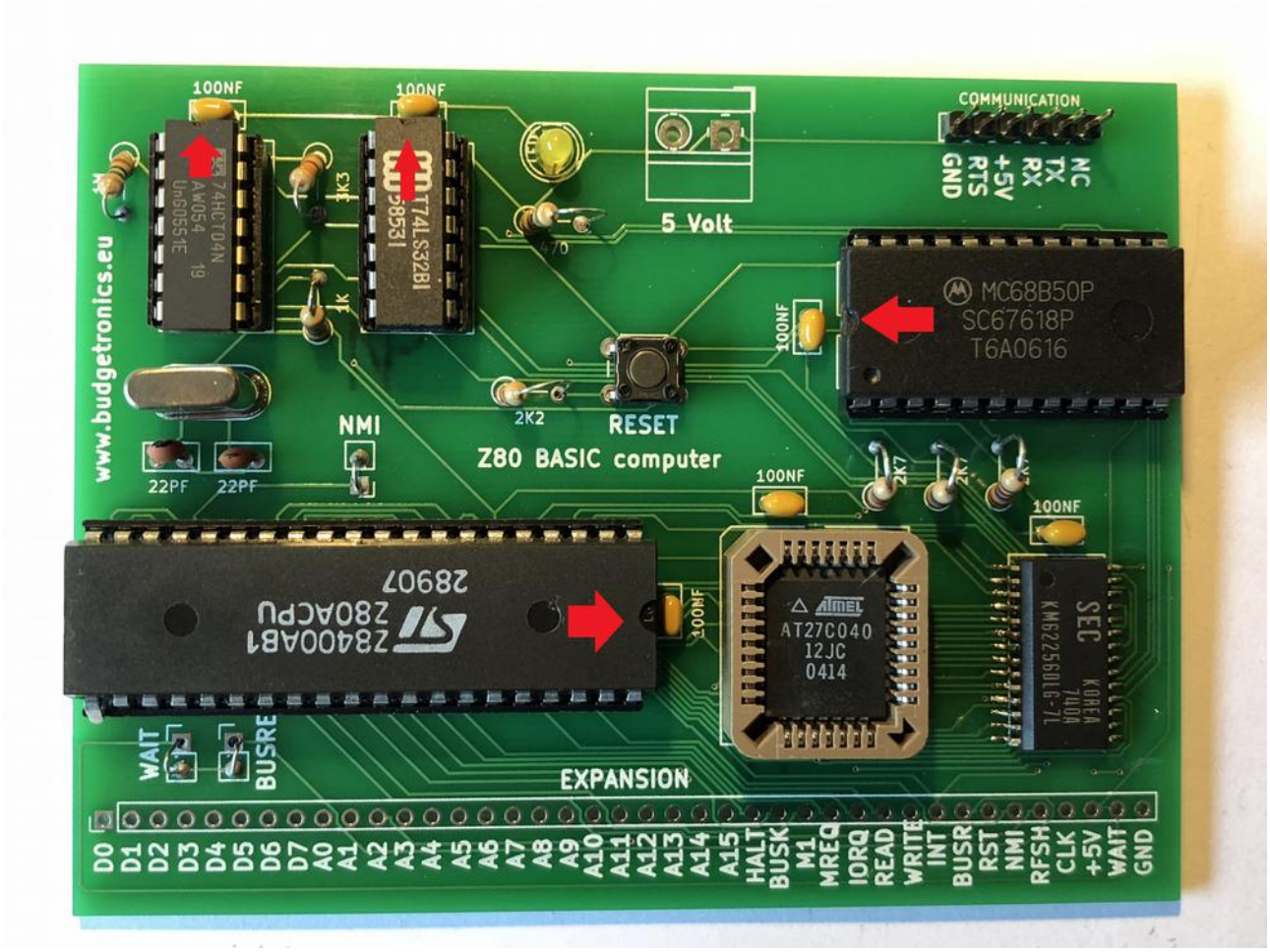


Put all 5 IC's in the right sockets

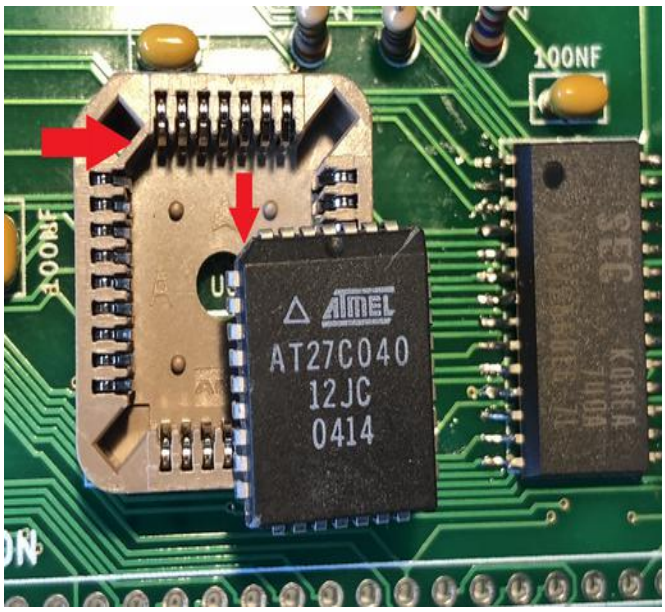


Take care that all the IC pins go straight in the sockets. Sometimes you have to bend the pins a little bit so they fit the socket. You can do this by placing one side of the IC against a hard surface and press so the pins will be aligned in a 90 degrees corner.
Also take care you place the IC's in the right way in the sockets. Look at the picture below.

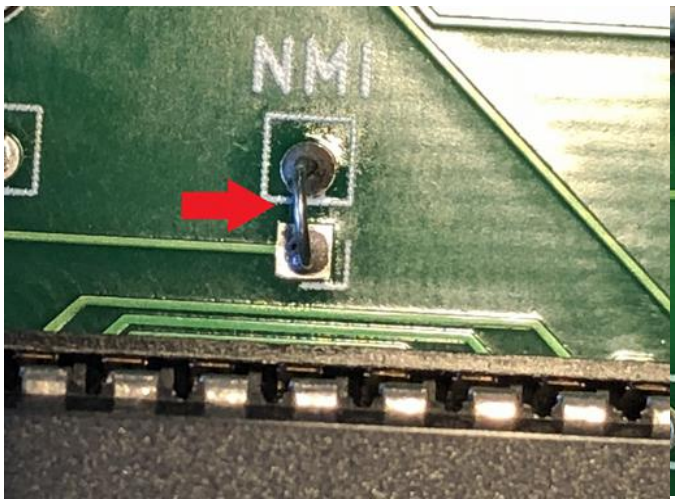
The red arrows show the side with the small notch. This notch is also visible on the PCB.

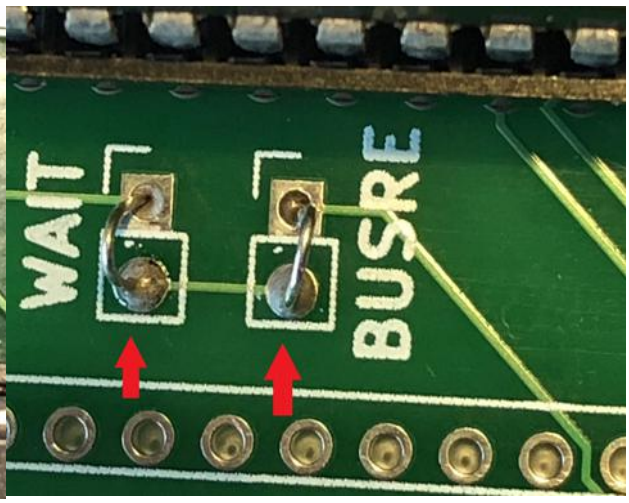


The last part is the placement of PLCC ROM IC. Push this flat in the socket all down. Watch the right orientation.

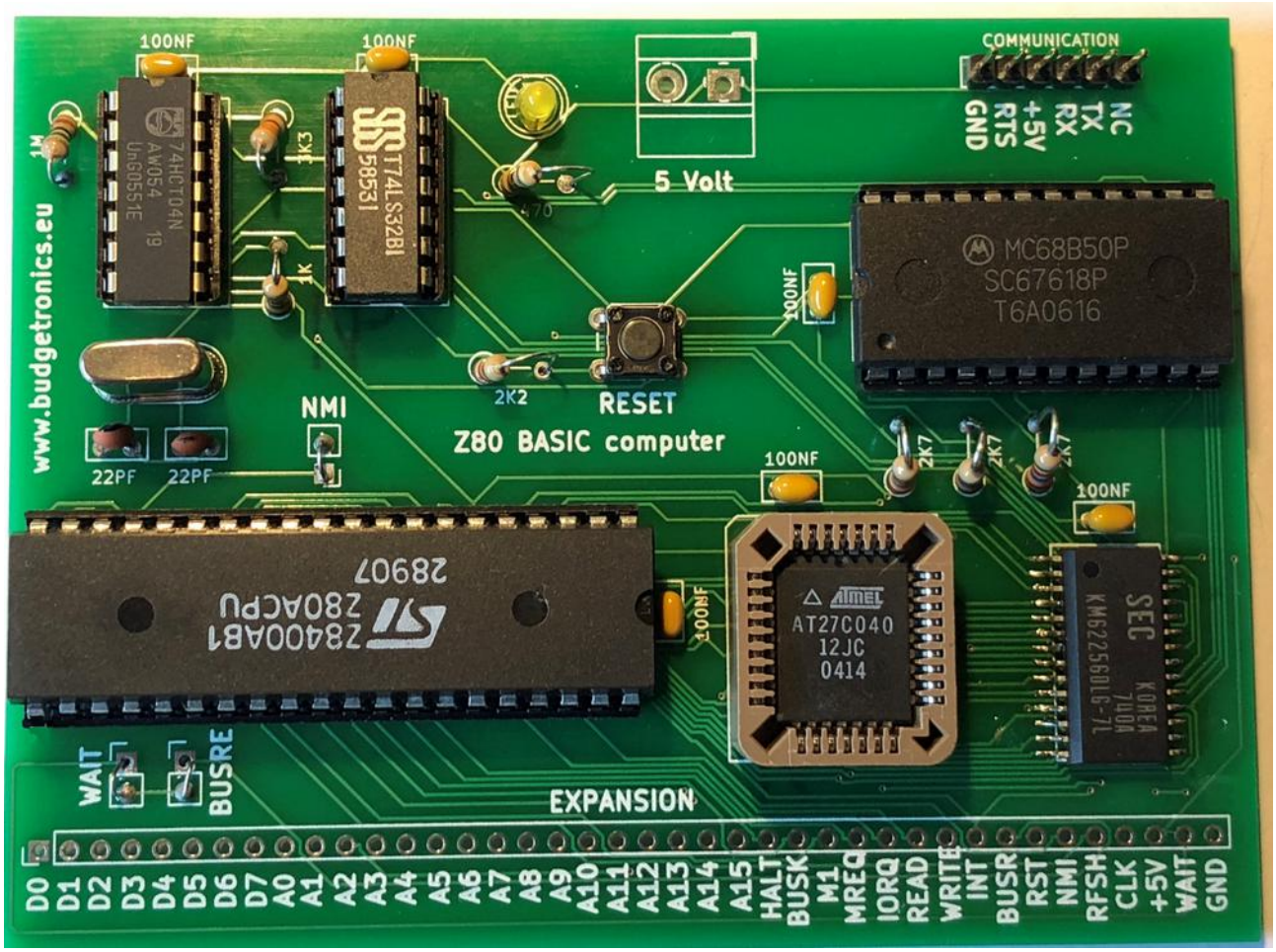


The last solder work is placing the 3 wire bridges on the PCB, NMI, WAIT and BUSREQ. You can use the cut off wires for this.





Now the computer is finished and it will look the same as in the picture below.



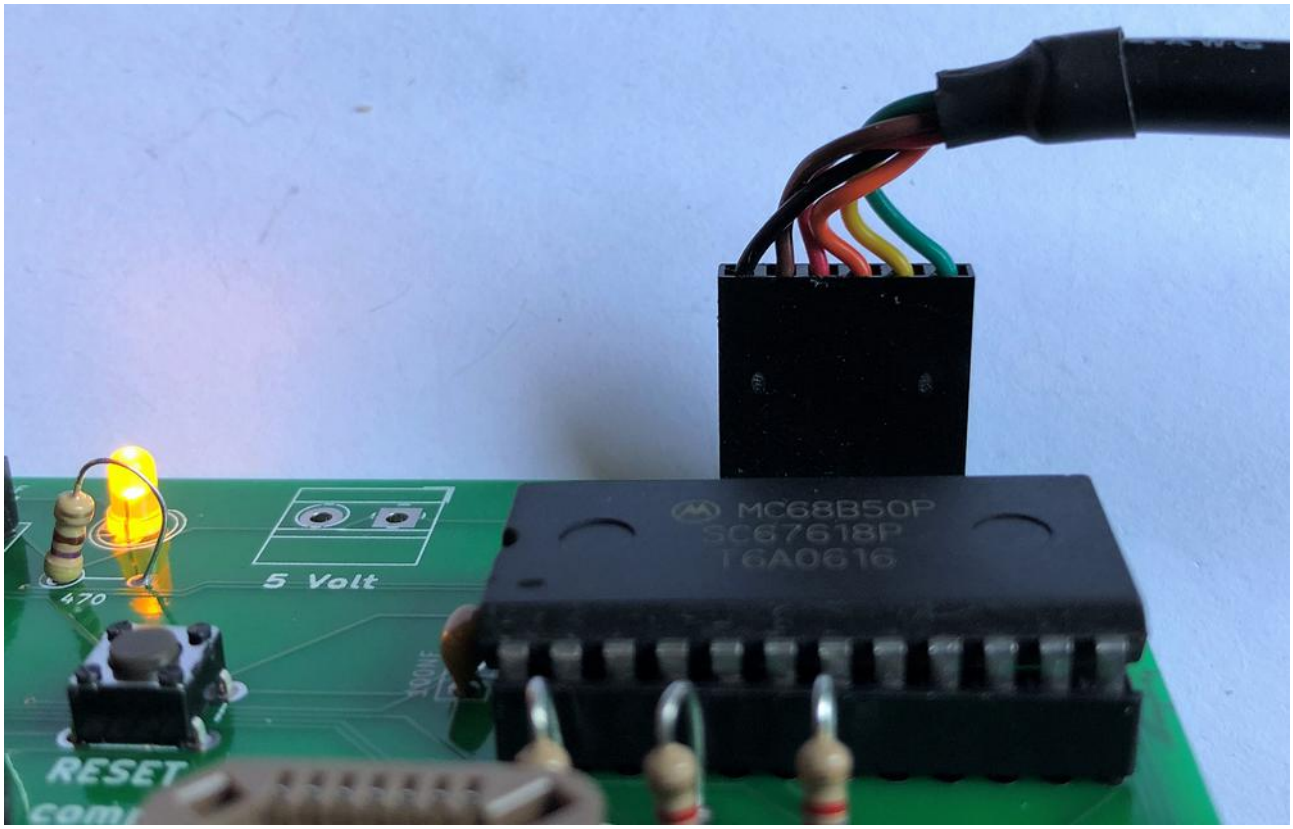
The 5 Volt connector is not connected here because normally the 5 Volts comes from the TTL to USB cable on the 6 pins header. Only if you have another solution you can use the 5 volts connector but NEVER connect them both to 5 volts.In a case like that use only the Tx, Rx, and if needed the RTS, connections on the header.

If you have a serial to USB cable as shown below the you have all the connections you need. The black wire (GND or -) is situated on the left side. The RED wire is + 5 Volts.

Yellow is here Rx and orange is the Tx connection. So this is exactly the opposite of what is indicated on the PCB but this is the right way to connect. You always have to connect the Rx to the Tx and theTx to the Rx of the serial connector. So crosswise. The other connections are just as indicated on the PCB.

If you connect the power the LED will light up.

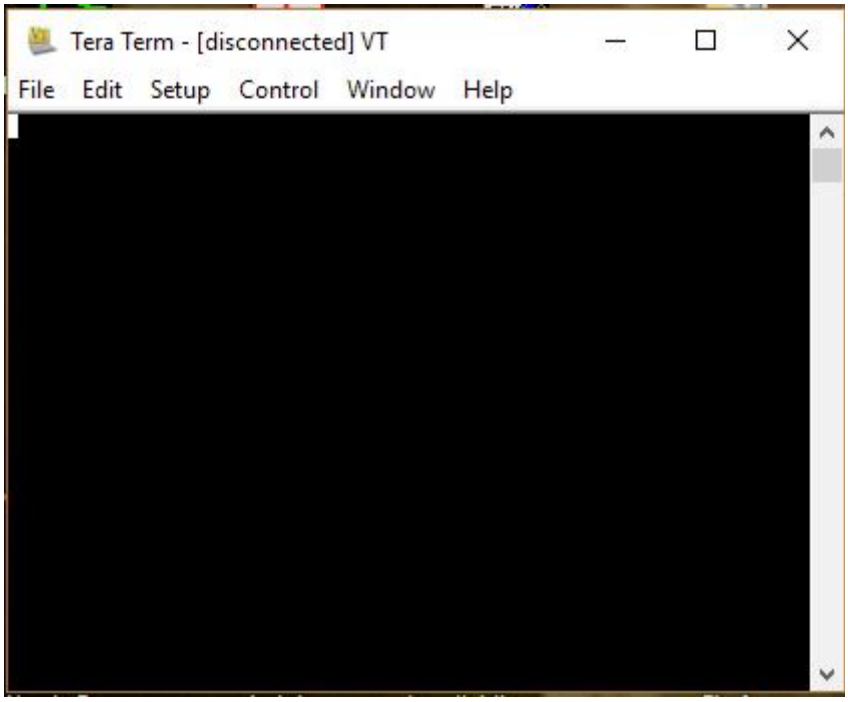
But before you do this check the PCB to see if you soldered in all the parts, made no solder mistakes and that all parts are orientated in the right way.



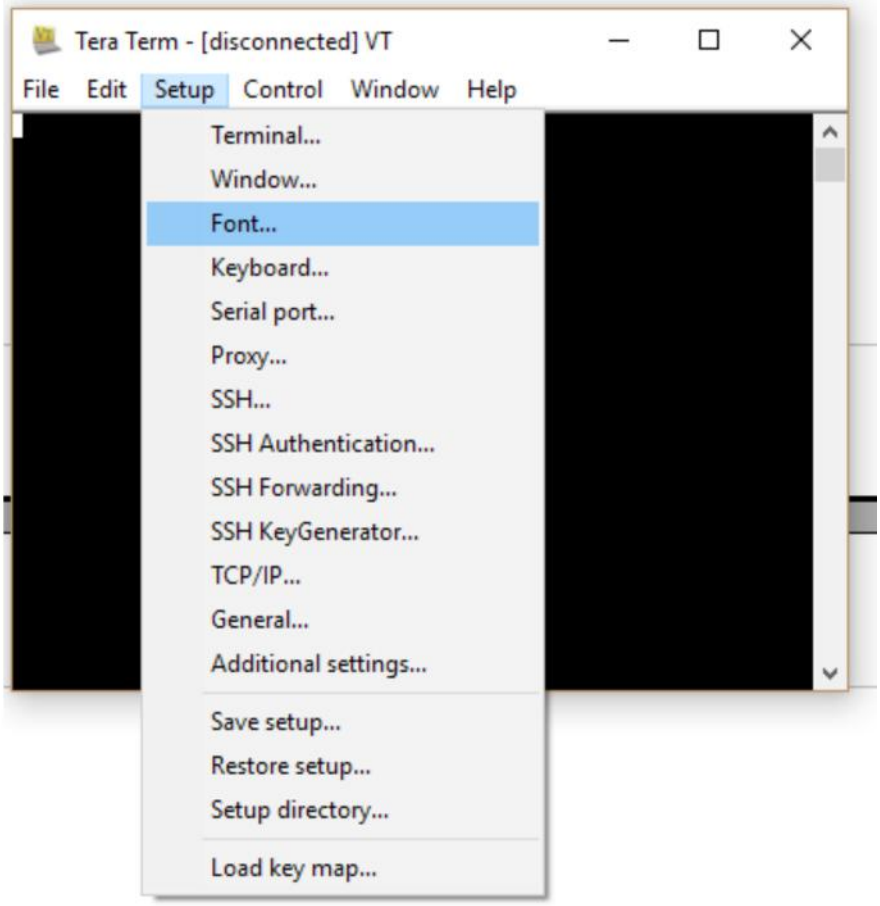
You always can choose another way to read out the serial data. We have only shown you one possible way to do this but there are more solutions. Maybe you still own a real terminal or have made another solution like the the ASCII video terminal building kit as mentioned earlier. In this way you add a VGA video output or composite video output and a PS/2 keyboard input to make this a stand alone computer you can add See menu Buidling kits in our website www.budgetronics.eu for more details about this kit. It is all up to you.

STARUP AND TESTING

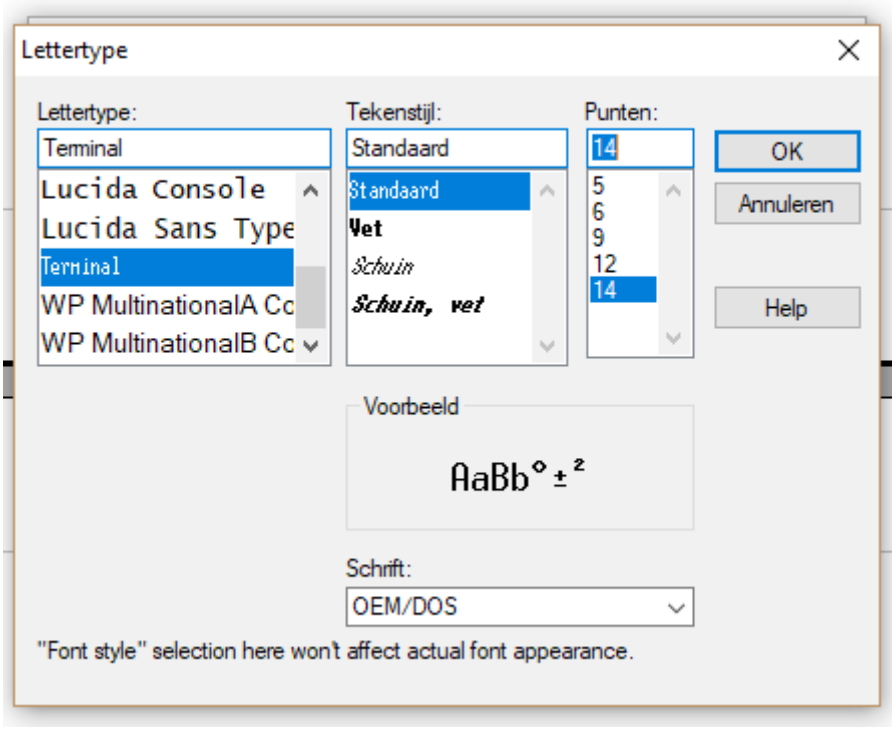
Start Teraterm.



Goto menu Setup/Font and click.

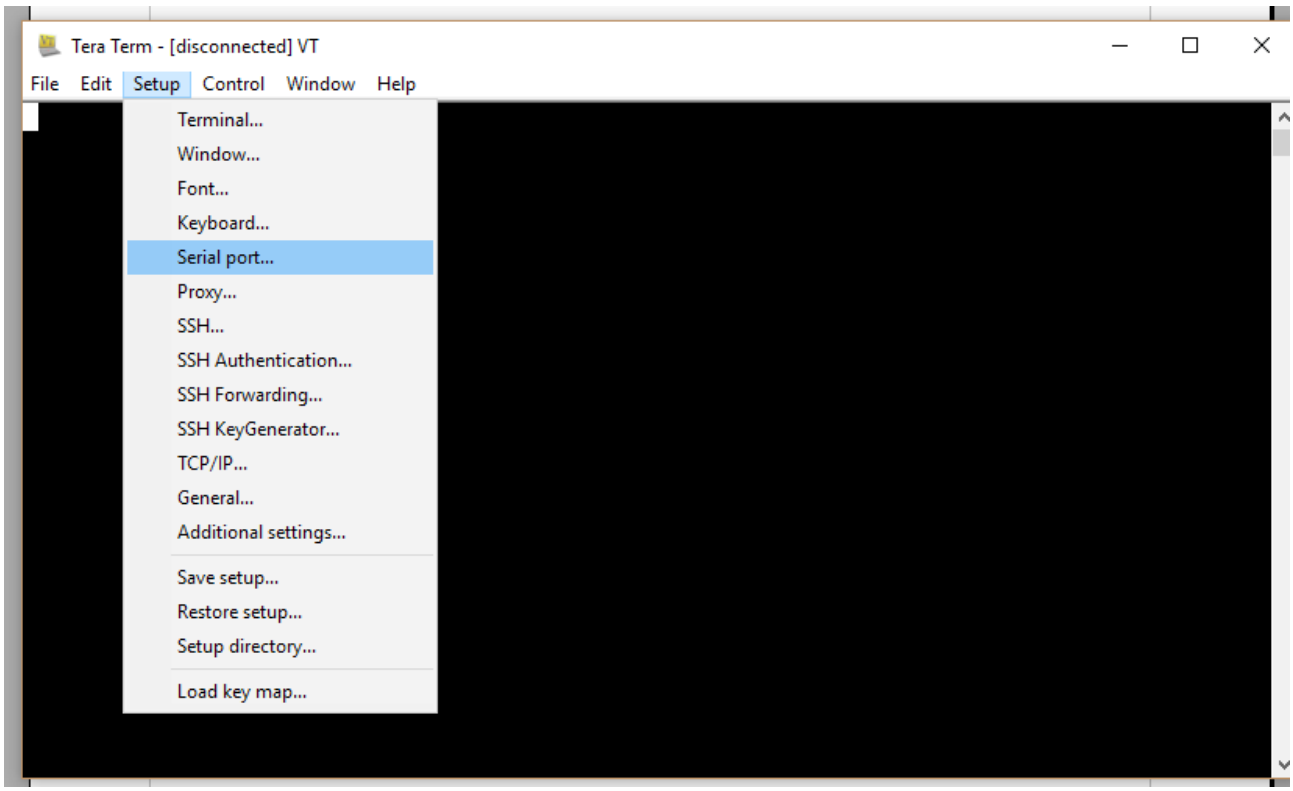


A window opens. Choose font 14 for a bigger more readable font format..



Connect the serial to USB cable between a PC and the Z80 retrocomputer and check that the LED is on as prove of power up.

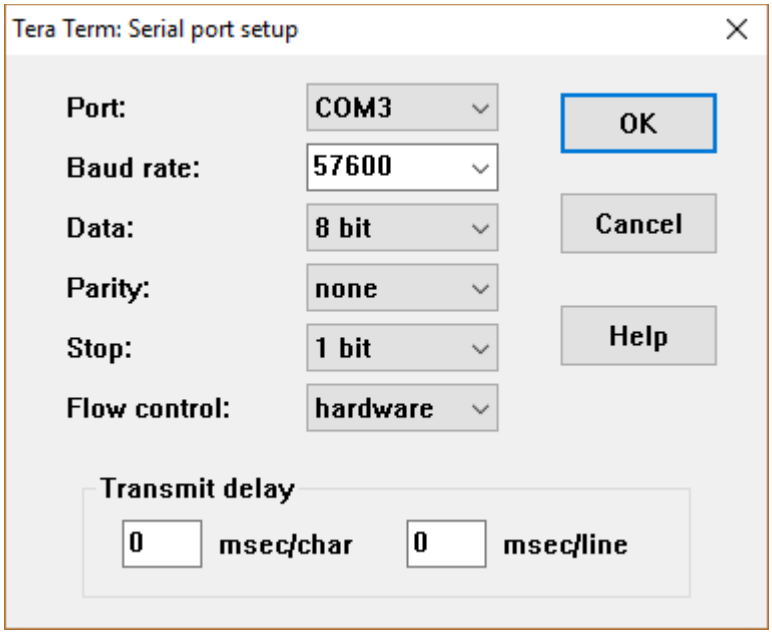
Click OK and goto menu Setup/Serial port



Type in baud rate 57600, Data 8 bit, Parity none, Stop 1 bit en Flow control hardware.
ATTENTION! If you have a serial to USB cable with only 4 connections +5V, GND, Rx, Tx and no RTS, you have to set flow control to **NONE** instead of hardware.

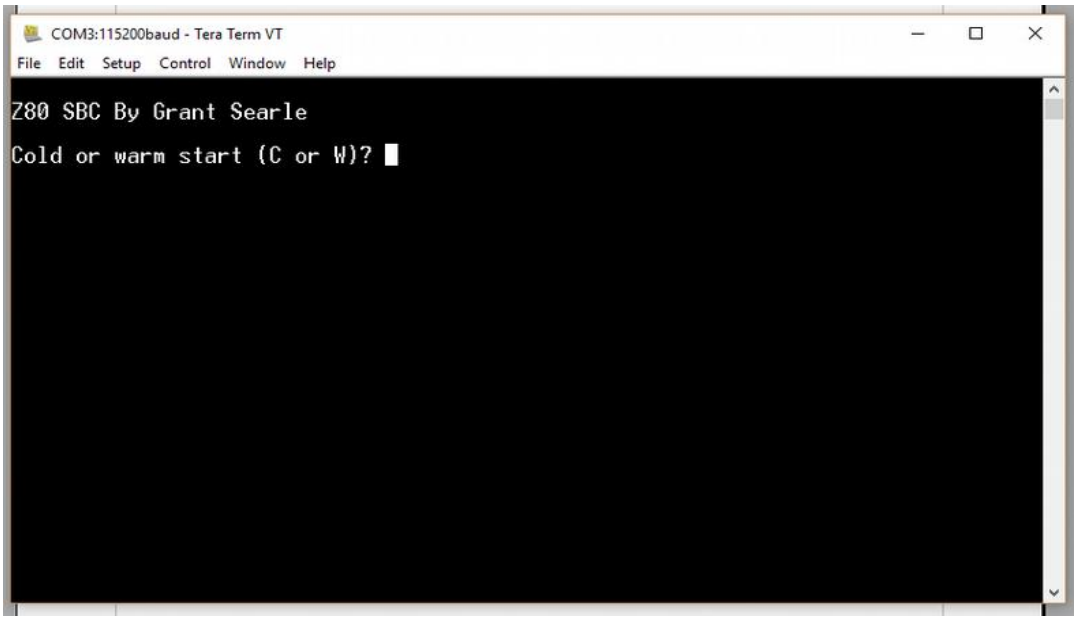
(If you want to try to run the Z80 CPU on 7,3728 Mhz the you have to replace the quartz crystal and put the Baud rate to15200, Remember this will not always work with every Z80 CPU because the supplied CPU is made for 4 mhz speed.)

If nothing appears in the “Port:” box the the Z80 computer is not connected to the PC or the USB to serial interface is not recognized. Check this. Below you see COM3 but this can be another COM port on your computer. As long as the PORT box shows a COM then there is a connection. Click OK and the setup is done.



If all is right the word “disconnected” at the top of the Teraterm window disappeared. If this is not the case go back to setup/serial port and press OK again.

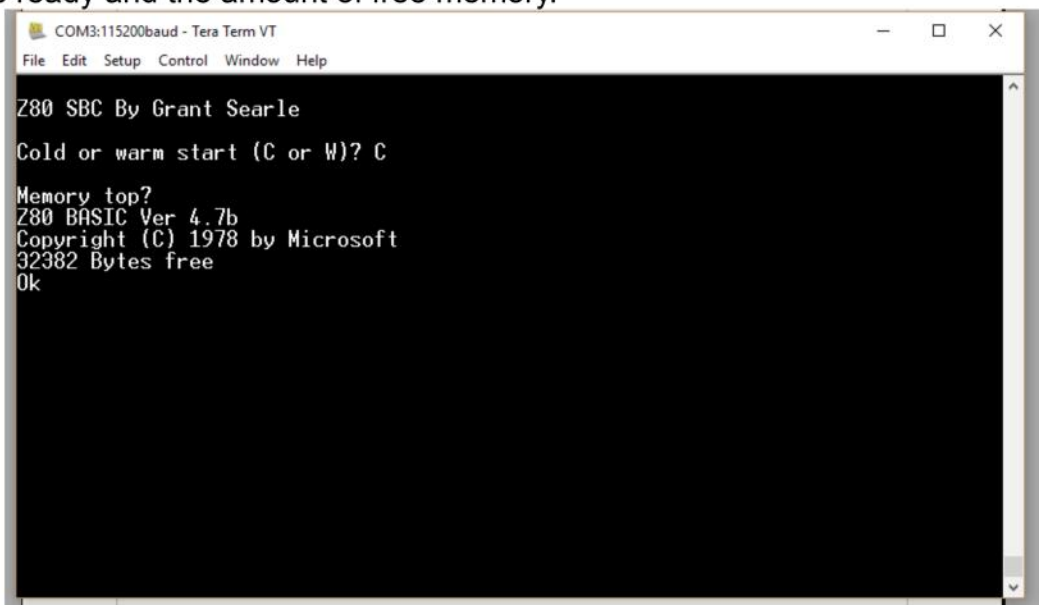
Press the reset button on the Z80 computer and if all is welll you will see:



If you get this message congratulations the computer works. If not go back check everything and read this manual again to find the fault.

The computer asks for Warm or Cold start. Choose C. The Warm start is only needed if there is a program in memory which you do not want to be deleted.

At the question memory top? Just press enter and you will see a startup message that BASIC is ready and the amount of free memory.



You now can begin programming the computer with the following commando's:

SGN,INT,ABS,USR,FRE,INP,POS,SQR,RND,LOG,EXP,COS,SIN,TAN,ATN,PEEK,DEEK,LEN,STR$,VAL,ASC,CHR$,LEFT$,RIGHT$,MID$

END,FOR,NEXT,DATA,INPUT,DIM,READ,LET,GOTO,RUN,IF,RESTORE,GOSUB,RETURN,REM,STOP,OUT,ON,NULL,WAIT,

DEF,POKE,DOKE,LINES,CLS,WIDTH,MONITOR,PRINT,CONT,LIST,CLEAR,NEW

TAB,TO,FN,SPC,THEN,NOT,STEP

+,-,*,/,^,AND,OR,>,<,=

Explanation about how to program in Basic go's to far for this manual. There are enough books available about this subject.

To test the computers Basic type in the following program: (After each line press ENTER):

```
10 A=0
20 PRINT A
30 A=A+1
40 GOTO 20
```

Type in LIST to see if the program looks exactly the same as shown here. If so type in RUN and the program will start and will count. To break out of the program pres Ctrl and C together.

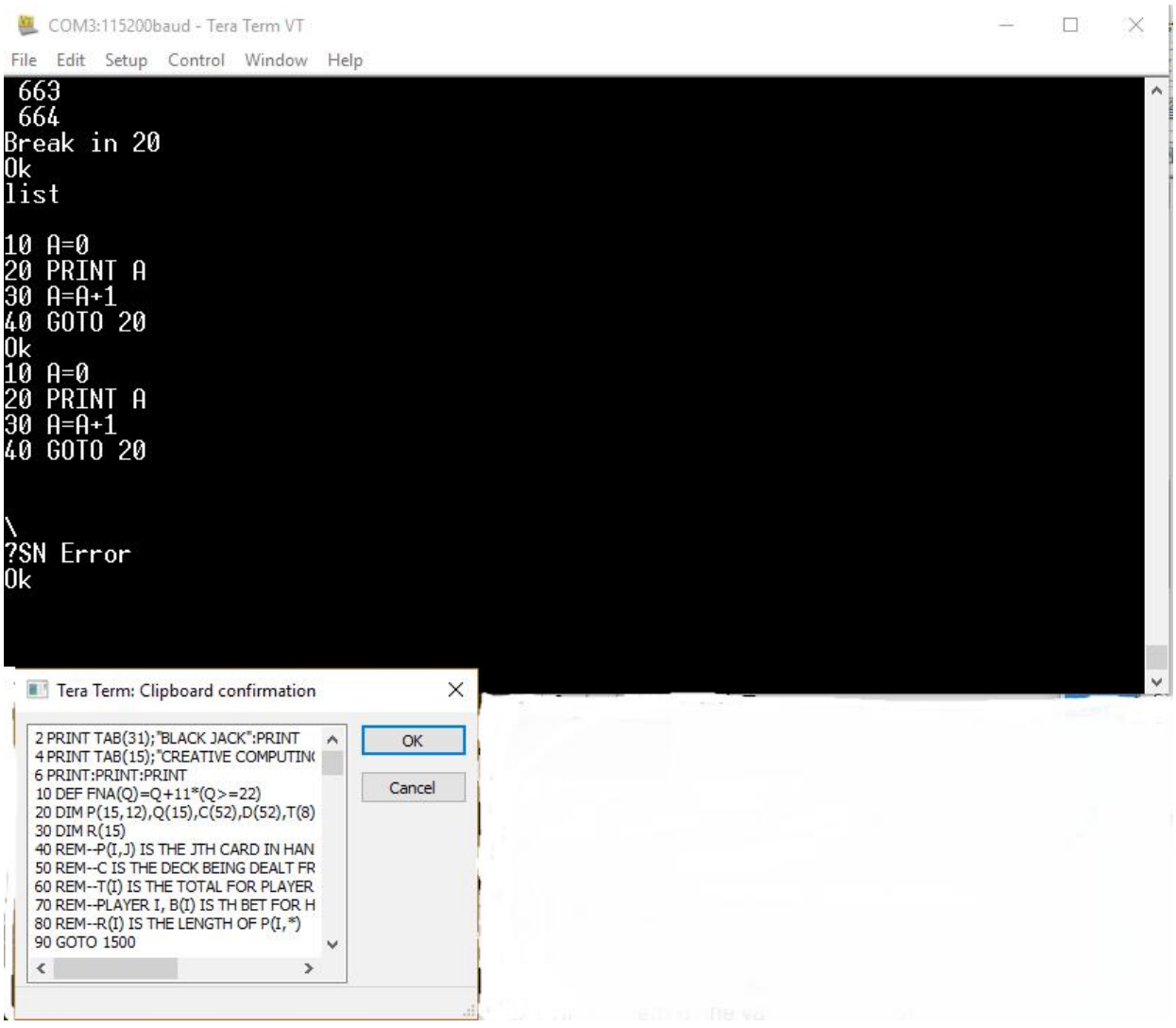
There is a small download package for this computer available on our website www.budgetronics.eu. Not only the terminal program Teraterm is in this package but also a nice collection of old Basic programs. Look on the left side, down, menu DOWNLOADS.

You can open the Basic listings with Notepad.

You can type in these listings but there is a faster way.

Open a programm in Notepad:

1. Click in the Notepad page.
2. Hold Ctrl A to select the whoe text.
3. Hold down righthand mouse button and select copy.
4. Click with the right hand mouse button in Tera Term window and a window pops up as shown below. Click on OK.



You will see the program is now transfered to the Z80 retrocomputer. After it finished you can type in the LIST command to see if it was transfered correctly. If so you can give the command RUN and you can play the game.

If you want the break in to the program press Ctrl C together and the computer will stop.

If you want to erase memory for another program just type in the command NEW.

It can happen that the program is not transfered correctly. This can happen when your USB to serial connection is delayed for some reason or if you do not use Flow control in the serial setup. This can be solved. In this case Goto menu Setup/serial port and change 0 in box “transmit delay” in say for example 10. You will see transfer is a lot slower but correct.

You can change the programs if you want or write your own but if you remove power all will be lost.

To preserve programs you can save them. In the Tera term window select the programcode and click the right hand mouse button. A pop up appears with the selected text inside. If you want you can paste the text in an empty Notepad page and save it.

This is the most simple method but you can also experiment with Tera Term menu File/Transfer ,Xmodem, Ymodem etc to send and receive. Try it out and experiment to learn.

If you add the ASCII VT100 terminal building kit to this computer, just connect the RX to the TX and the TX to the RX connection (crosswise) and also connect the GND connections of both modules together, supply the Z80 computer and the ASCII video module with 5 volts power, use the setup menu of the ASCII terminal to adjust the serial setup to the right specs, as described above, and you are ready to go.

Simple Z80 monitor from Phil green

In the Rom we added the simpe Z80 monitor which Phil Green added to the Grant Searl Rom. The below text is written by Phil Green to explain the monitor.

Monitor Commands:

Selecting **'M'** from the signon message will place you in the monitor and you will see the "Mon >" prompt, ready to accept commands. There is no 'backspace' correction, all arguments are taken as the last four digits typed before pressing Return. If you make a mistake, just continue typing with the four correct digits, ie 99579958 registers as 9958.

A is the hex arithmetic function. Given two arguments x & y it will display the sum x+y, the difference x-y, and the relative jump offset necessary to jump from address 'x' to address 'y''. Note that I reversed the subtract parameters with respect to the Nas-Sys arithmetic command as y-x always seemed the wrong way around to me!

Syntax:
Mon >A
3456 <return>
3434 <return>
SUM DIFF RJ
688A 0022 DC
Mon >

B is the 'Return to BASIC' command. It takes no arguments, just type B and the signon message will appear, just as though you'd pressed a hardware reset.

C is the copy function. Its intelligent so it doesnt matter if the source and destination areas

overlap.

The syntax is:
Mon >C
9000 <return> "copy from address 9000"
9800 <return> "...to address 9800"
80 <return>"...for 80 hex bytes, ie 9000-9080 will be copied to 9800-9880"

D is the display command. It takes one argument which is the start address, and displays in hex and ASCII eight lines of sixteen bytes of code.
The syntax is:
Mon >D9000 <return>

G is the "Go to" command which takes one argument as the start address and will run a user program from that address.
Syntax is:
Mon >G9000 <return>"Go to 9000 hex and start executing code there"
If you unintentionally enter “G” the goto can be aborted with a full-stop “.”
Programs can return to the monitor using either a RET instruction or a jump to 1DC0H.

I is the Intel-Hexloader command. It doesnt take an argument as the load address is specified within the hex file. Once 'I ' has been pressed, begin sending the file (at fast baud rates you may need a line-delay of maybe 100ms or so). For every line received, a dot is displayed to show progress. Unlike many implementations, this is a fully error-checked loader - non-hex characters and bad checksums will be identified with "~Hex!" and "Csum!" messages. You can see loading progress on the CH340G LED.

M is the modify memory function. It takes one argument as a start address, and displays the current contents of that memory address. You may press or to step through without making changes, or type a new hex byte of two characters followed b . The command is terminated with a '.' full-stop which will bring you back to the "Mon >" prompt.
Syntax is:
Mon >M9000 <return>|
9000 54 <pressing return leaves content as 54>
9001 76 77 <keying '77' & return will change the contents of location 9001 to 76>
9002 45 . <a full-stop finsishes the modify session>
Mon >

All arguments are taken as the last four digits typed before pressing <return>. If you make a mistake, just continue typing with the four correct digits. For example, 93457281 is actually taken as "7281". If you are entering a byte (as when using modify) the same applies - if you typed 32 when you actually meant 23, just continue with the '23' because '3223' is taken as '23' in this context.

Single arguments as in Display and Modify are entered against the command letter, on the same line. For clarity, multiple-argument commands like Copy and Arithmetic place each argument on a separate line.

Have a lot of fun with the Z80 Retrocomputer!

One of our customers made a remark on how to try out Z80 machine code. His comments are included below for everyone to use.

Hello,

I recently bought and built your small Z80 Retrocomputer kit, which works absolutely fine. The construction manual is easy to follow.

You have a small section in the manual describing how to get started with BASIC programming.
If you want to start trying out Z80 machine code, there is one important fact that I could not find in the documentation.

The address of the vector for the USR() function, which from BASIC is used to call machine code.

I found out by disassembly that the addresses 8049h and 804Ah hold the lower and upper bytes of the address of the machine code called by USR().
So if one wants to call some Z80 machine code, you must essentially do something like this:

```
100 REM U IS ADDRESS OF MACHINE CODE TO CALL
110 REM AS AN EXAMPLE, USE COLD START BASIC 0156h
120 U=&H0156
130 REM V IS ADDRESS OF VECTOR FOR USR() FUNCTION
140 V=&H8049
150 REM STORE THE ADDRESS IN THE VECTOR
160 DOKE V,U
170 INPUT "PRESS ENTER TO COLD START"; A$
180 X=USR(X)
```

*Perhaps the documentation could mention that the jump vector address for the **USR()** function is located at addresses **8049h** and **804Ah**.*
If you want to put the above example BASIC program in the software download ZIP file, please feel free to do so.