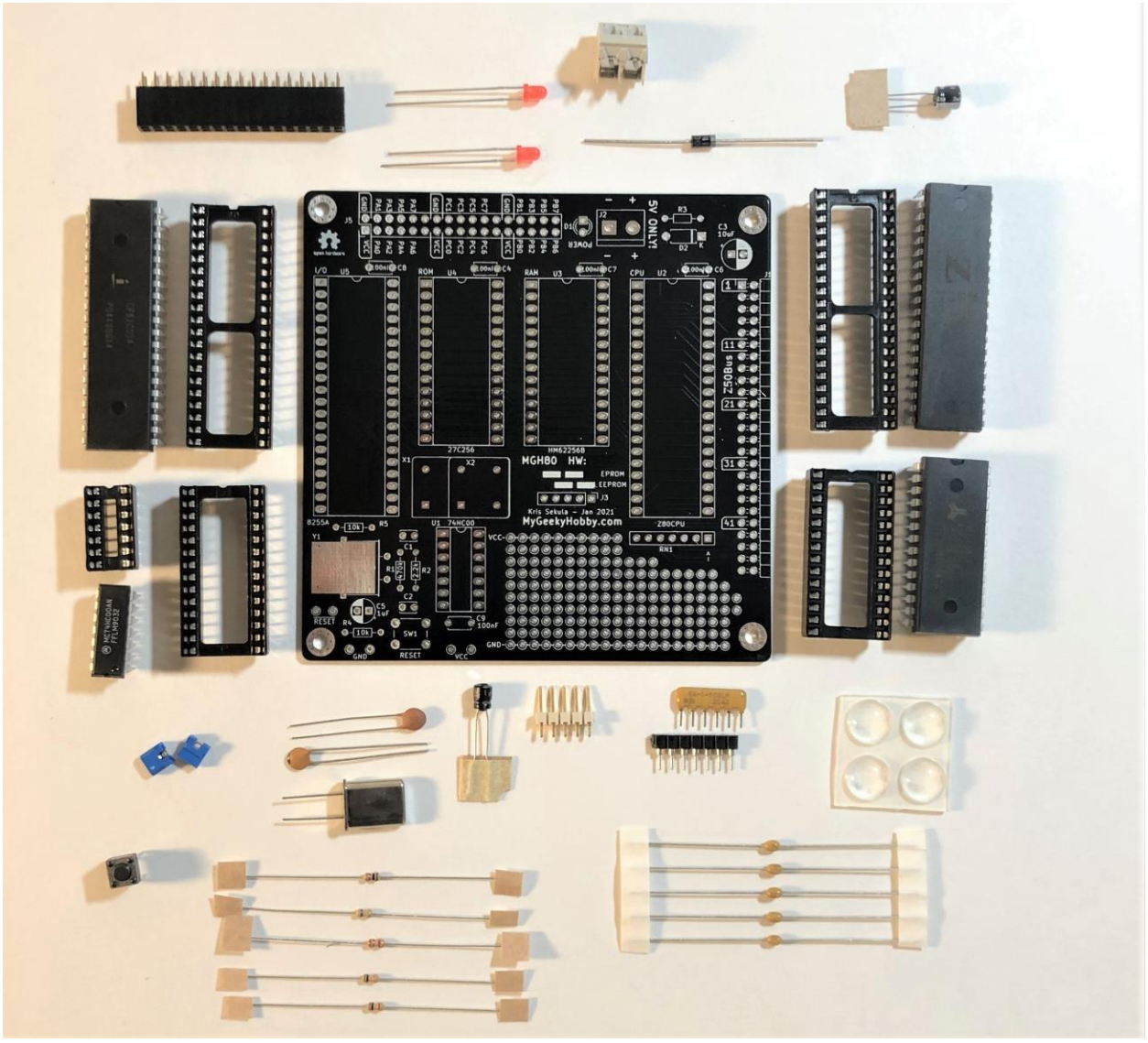
“MGH80” Z80 controller DIY kit – Assembly Instructions.



MyGeekyHobby.com ver 1.5.1 – October 2021

Thank you for purchasing the “MGH80” Z80 Controller DIY Kit, following is a brief assembly instruction.

This document is being updated regularly, check my GitHub page for updates:

https://github.com/Kris-Sekula/mgh80

BOM mgh80 v1.5

Qty	Reference(s)	Value
3	reset,gnd,vcc	wire jumpers (optional)
1	PCB	v1.5
5	C4, C6, C7, C8, C9	100nF
1	R1	470k
1	R2	2.2k
1	R3, R4	1k
1	D2	1N4002
1	R5	10k
1	R6	10k (optional)
1	RN1	SIP7 (F) socket
2	U2,U3	DIP40
2	U3,U4	DIP28
1	U1	DIP14
1	D1, D2	LED
1	Y1	4MHz
2	C1, C2	22pF
1	C3	22uF
1	C5	1uF
1	SW1	RESET
1	JP1,JP2	2x 2pin jumper for J3
1	J3	SIP5 (M) header
1	J5	34 pin header
1	J2	Screw Terminal
1	J1	50 pin header (optional)
4	H1, H2, H3, H4	Rubber feet
1	RN1	10k Resistor network
1	U1	74HC00
1	U2	Z80CPU
1	U3	32k SRAM (62256)
1	U5	82C55A

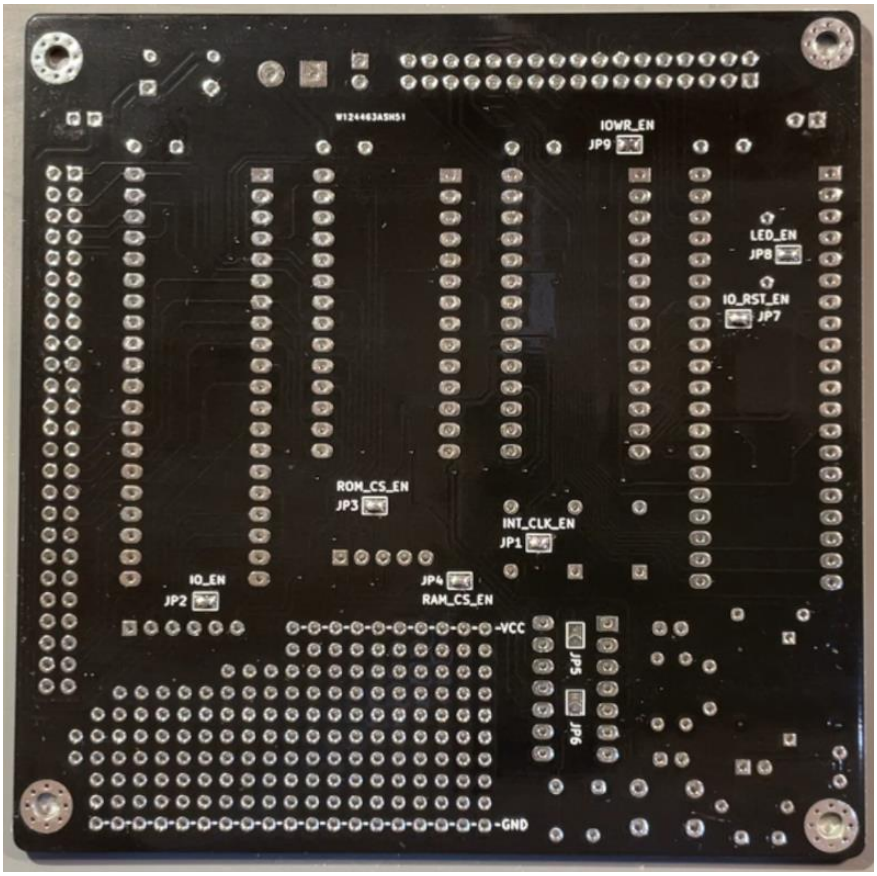
1. Configure the board option jumpers at the bottom of the PCB.

Memory/IO Decoder	Jumper	Standalone	With Serial Expansion
Enable internal IO Decoder	JP2	Closed	Opened
Enable internl ROM Decoder	JP3	Closed	Closed
Enable internal RAM Decoder	JP4	Closed	Closed
Enable internal IO_WR	JP9	Closed	Opened

Other		
Reset circuit for 8255	JP7	Closed
Enable onboard LED on PA0	JP8	Closed

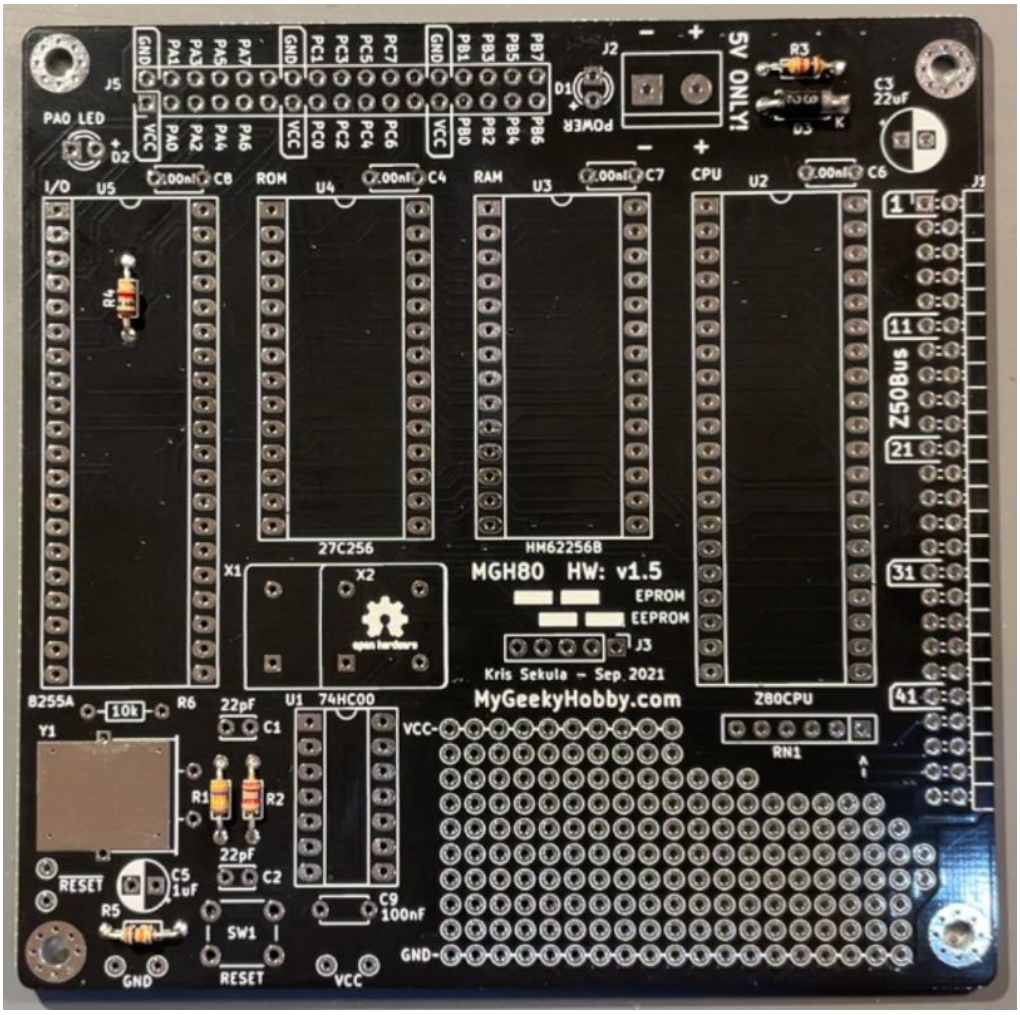
Clock source		Crystal Y1	X1/X2 module
Enable internal clock generator	JP1	Closed	Opened
Disable unused gate	JP5	Opened	Closed
Disable unused gate	JP6	Opened	Closed

Basic MGH80 should be configured as follows: using a “blob” of solder close **JP1, JP2, JP3, JP4** and **JP7, JP8, JP9**, but **JP5 and JP6 are** should be left **opened.**

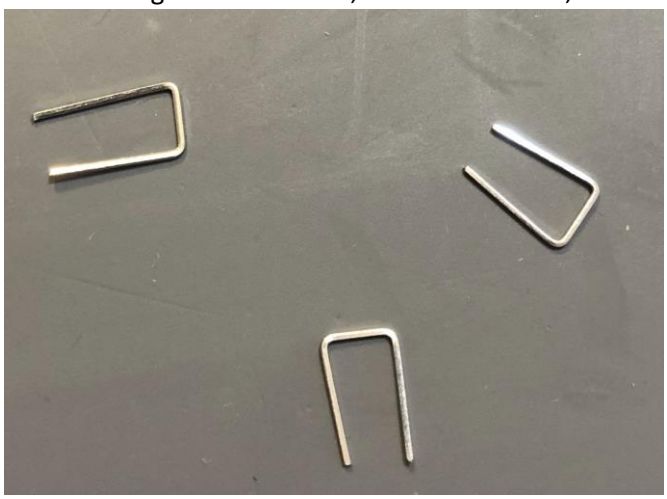


Note: If you are also using the “Serial Expansion board”, leave **JP2** and **JP9** opened.

2. Start populating components from the shortest to the tallest: R3,R4(1k), R1(470k), R2(2k2), R5(10k)…diode D3 (1N4001) pay attention to D3 polarity (white band):

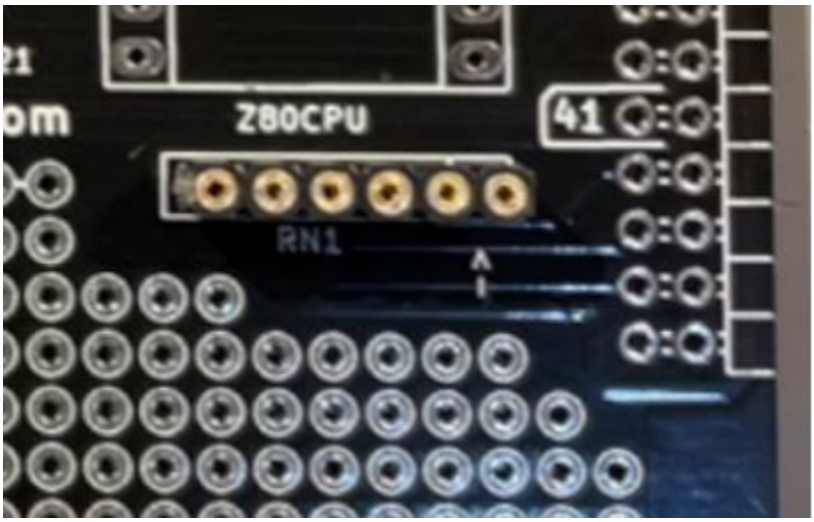


3. Using leftover wire from trimming of the resistors, create the RESET, GND and VCC test points:

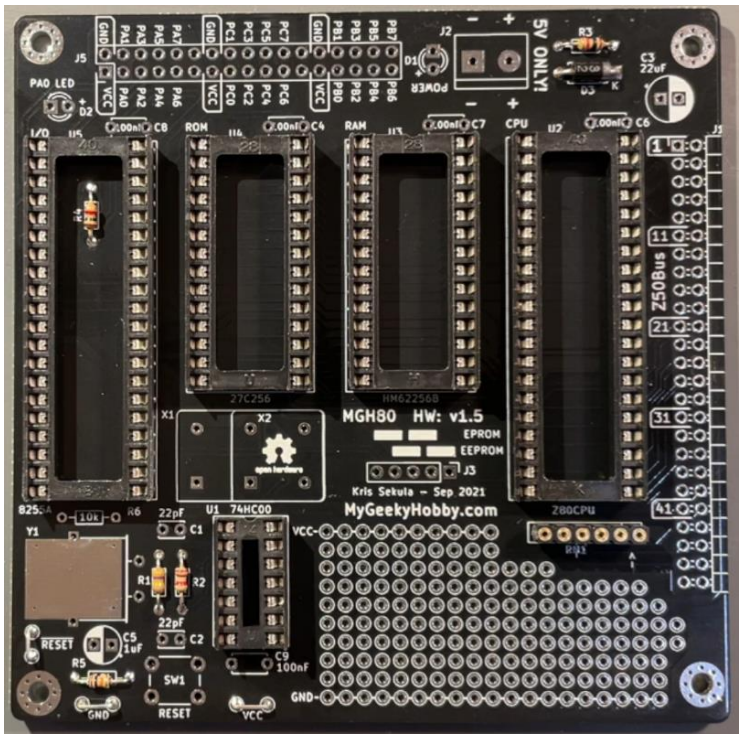


4. Solder the test points:

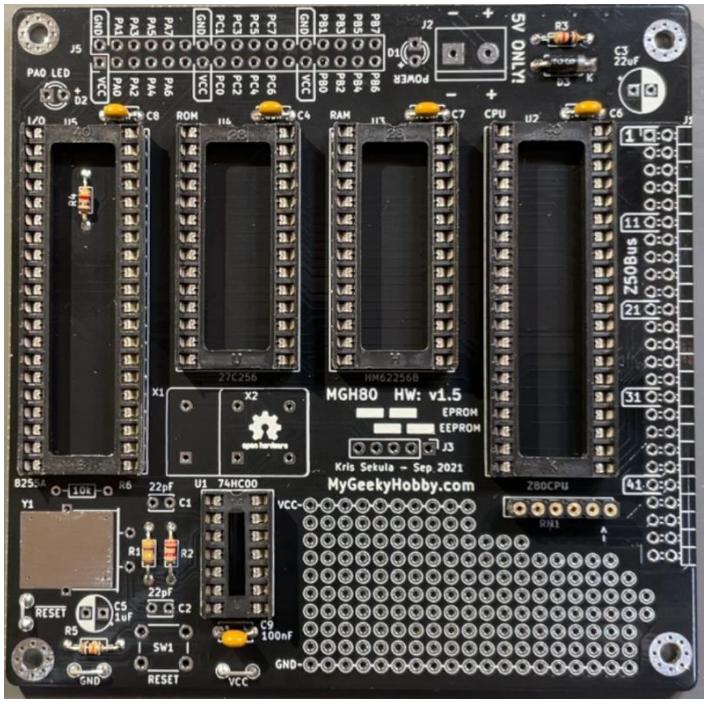
5. Solder the precision socket for resistor network RN1:



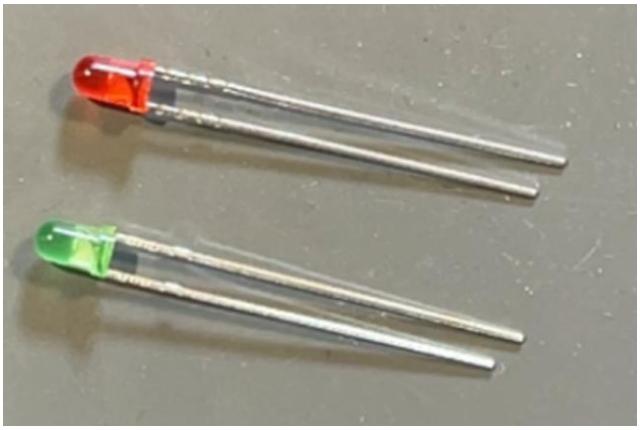
6. Solder IC Sockets (2x DIP28, 2x DIP40, 1x DIP14):

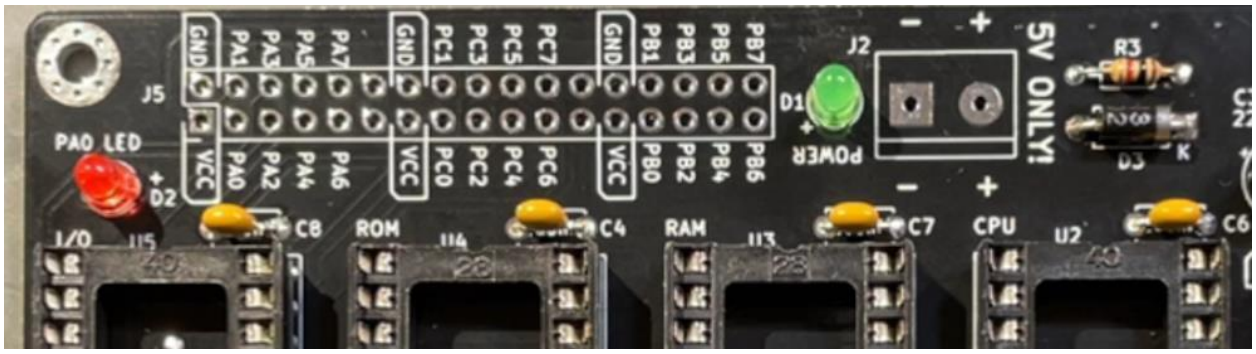


7. Solder 5 decoupling capacitors C4, C6, C7, C8, C9 (100nF)

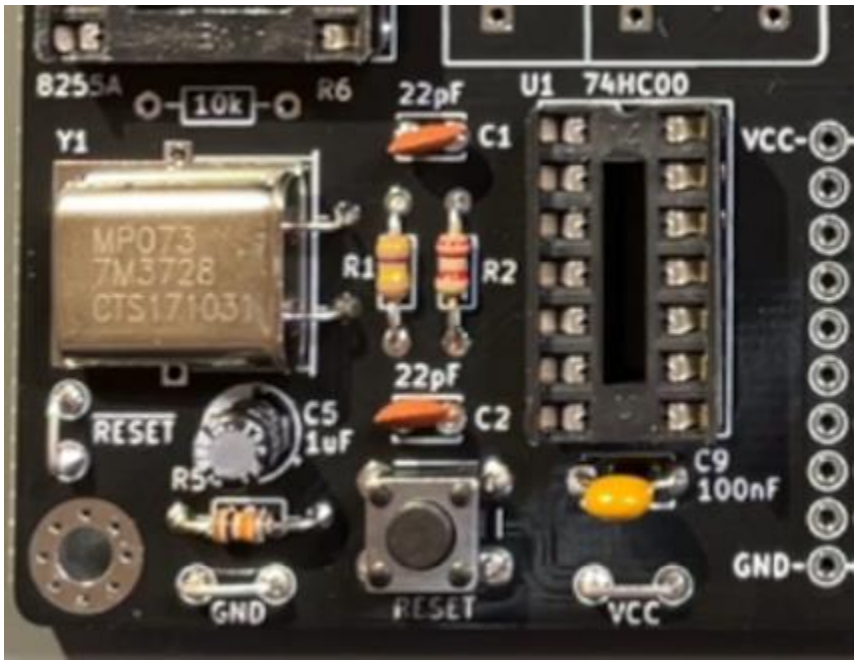


8. Solder the LED D1 and D2, the positive lead of the LED (Anode) is usually longer and it should be facing the “+” writing on the PCB:





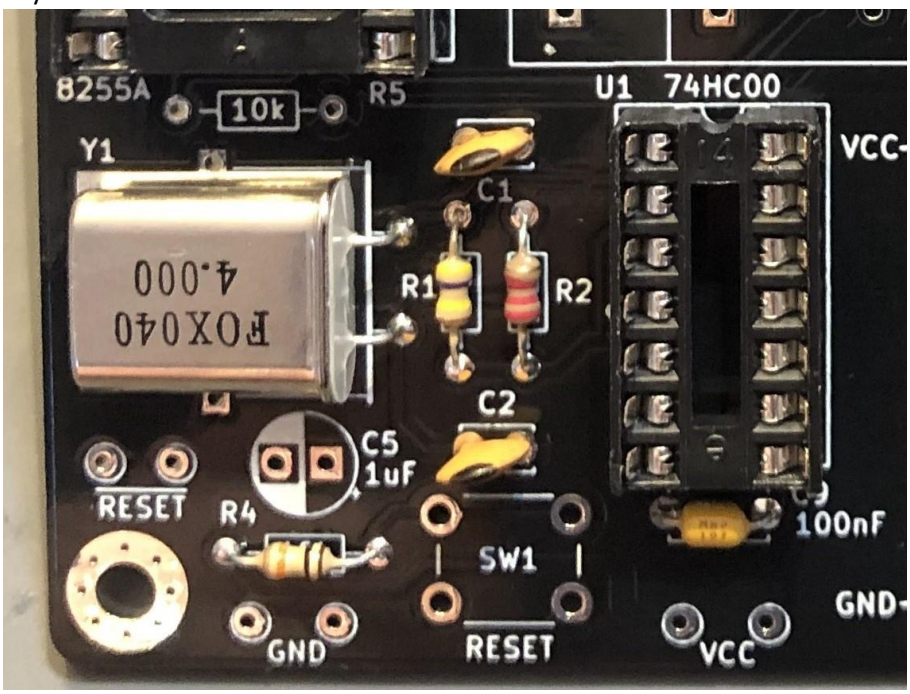
9. Solder the capacitor C1 and C2 – both are (22pF) and reset switch:



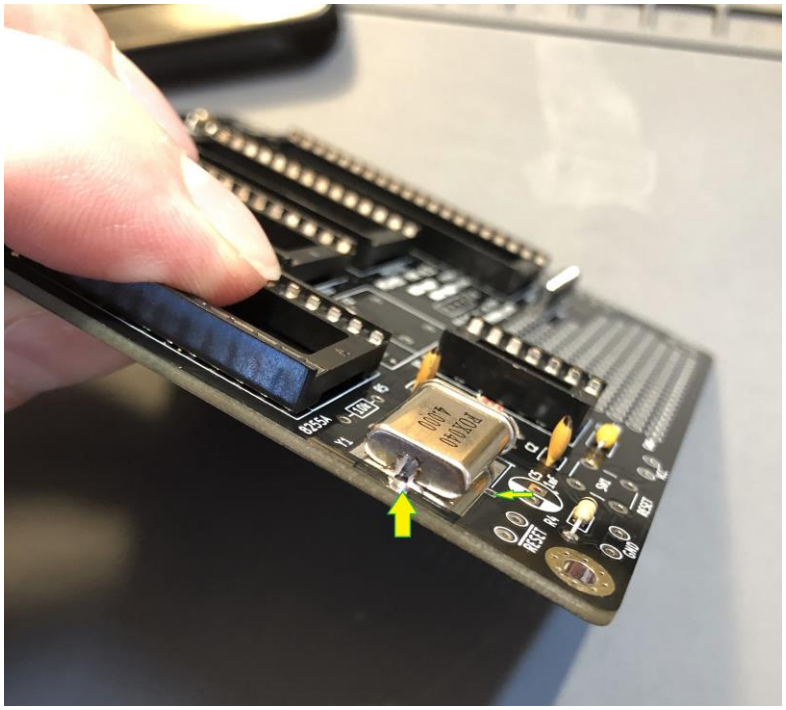
10. Prepare crystal oscillator Y1 by bending it's legs 90deg, I use the edge of the PCB to help:

Note: pictured is a 4MHz crystal, but if you are using the Serial Expansion board, you will need a 7.3728MHz crystal.

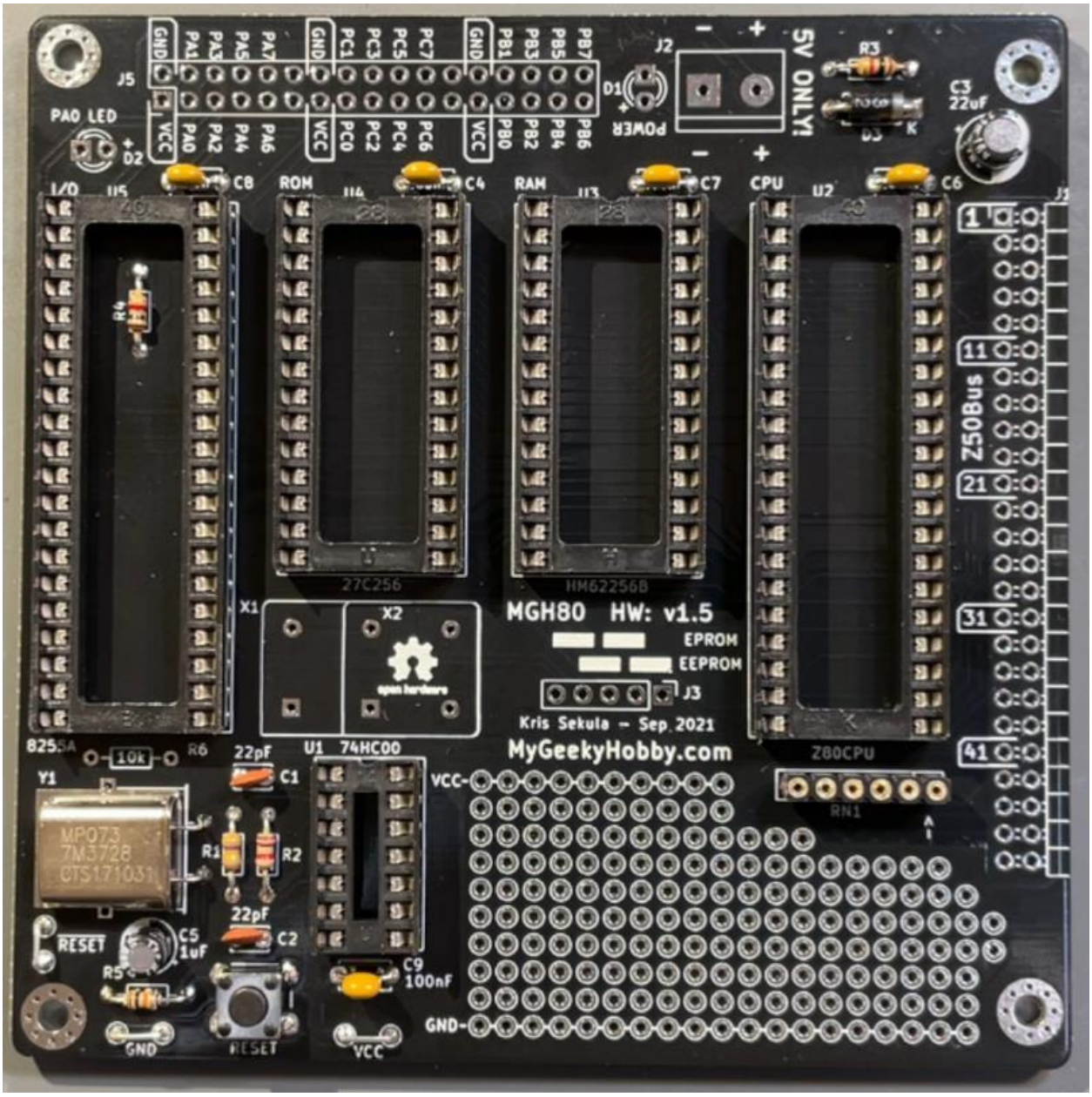
11. Solder the crystal oscillator Y1:



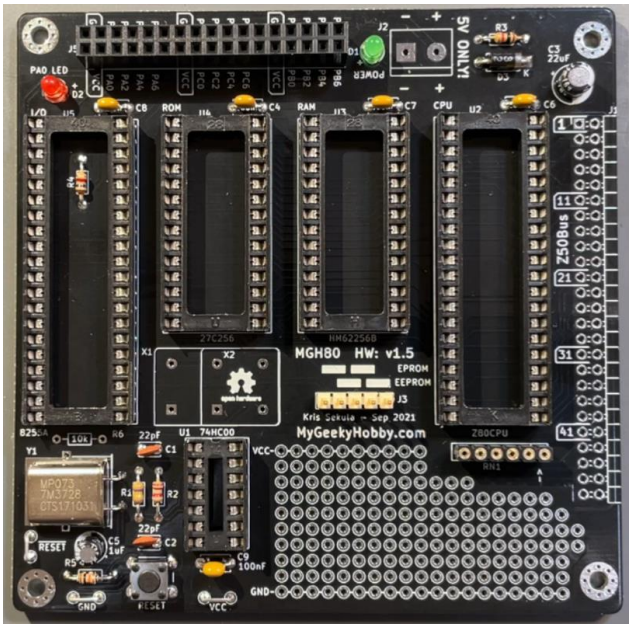
12. Secure the oscillator Y1 to the PCB with either a short link of wire or by soldering the oscillator directly to the PCB:



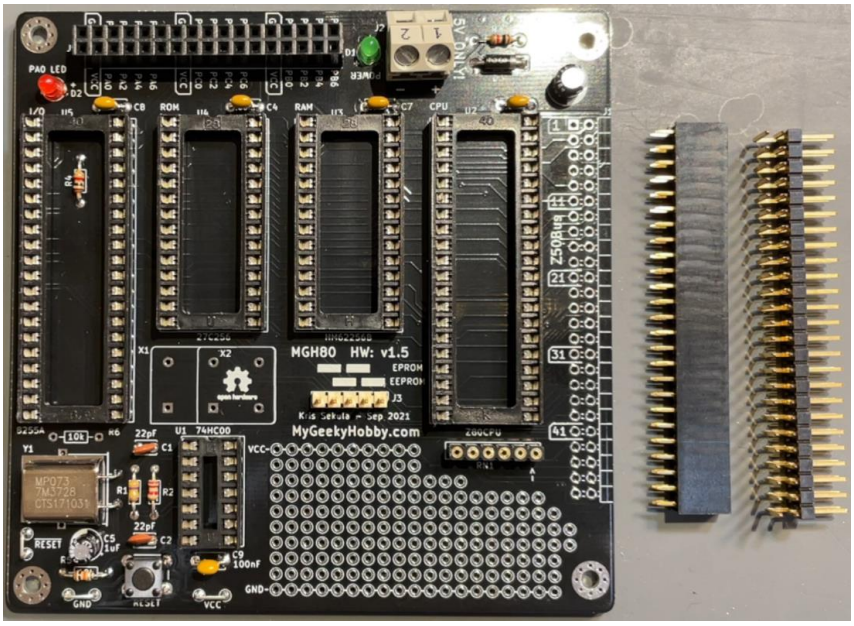
13. Install the electrolytic capacitors C3 (22uF) and C5 (1uF), be careful about polarity, white band on the capacitor marks "-".



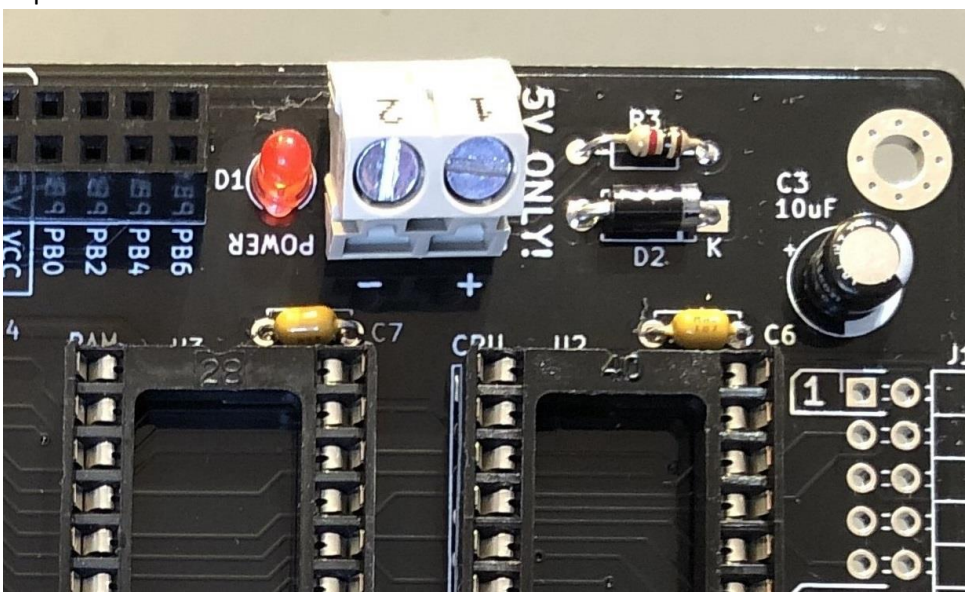
14. Install the 34pin female connector J5 and 5 pin male header J3



15. (optional) If you are planning to use the expansion board, install the z50bus connector, for "stacking" use 50 pin female header, for "bus" expander, use a 90deg bent 50pin male header:

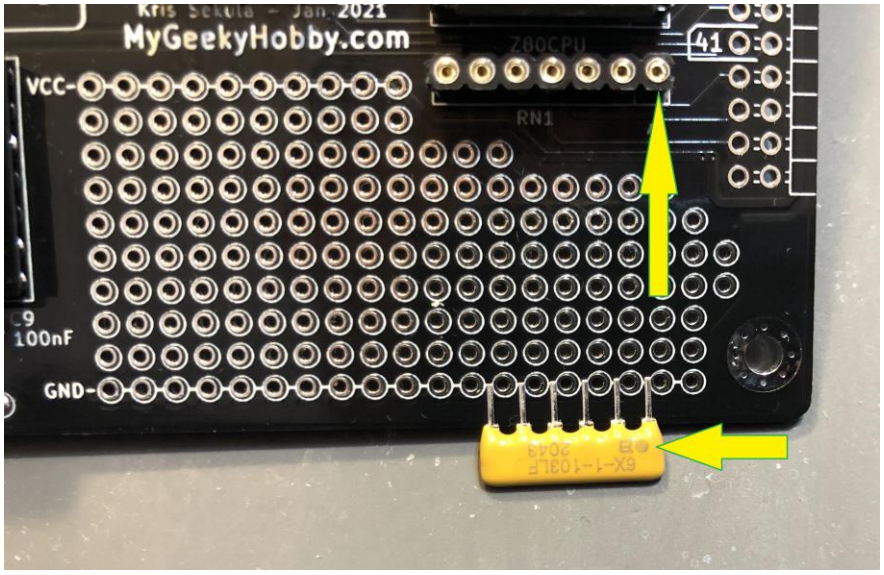


16. Install the power connector J2:

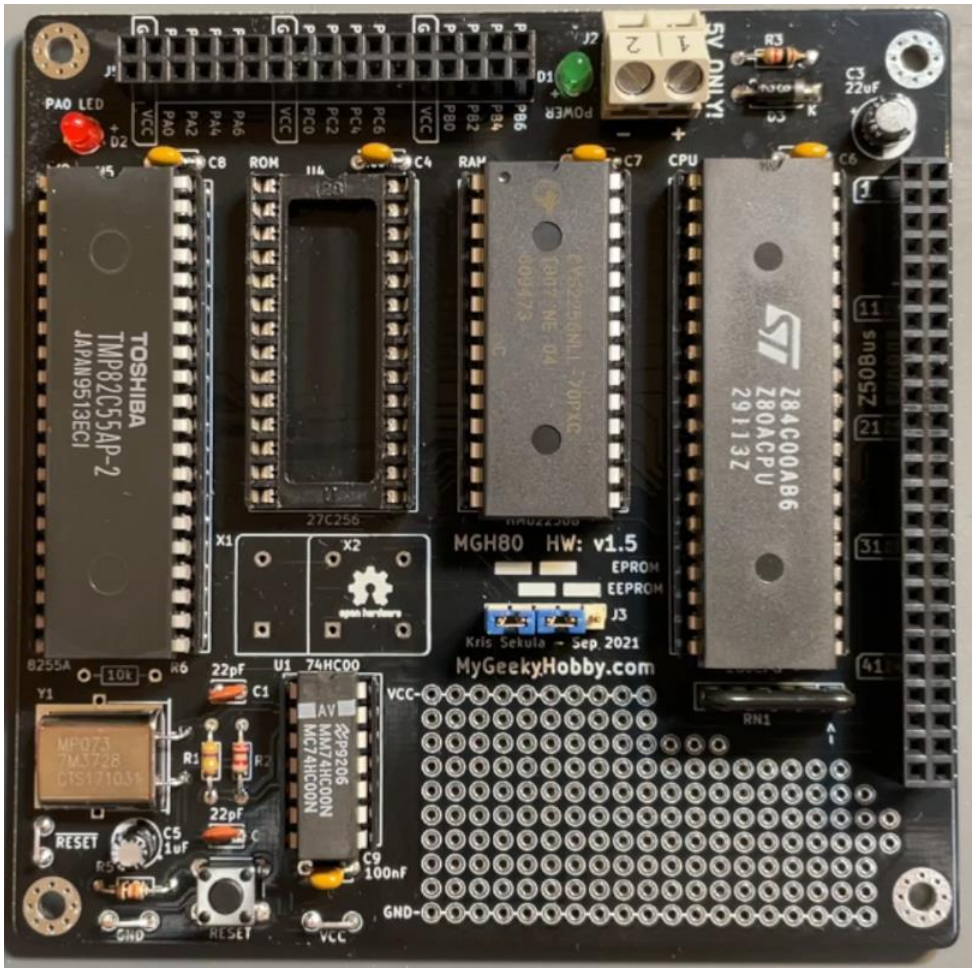


At this point all components are soldered, you should clean the solder residue with an appropriate cleaner (Isopropyl alcohol or similar). You can now insert components into sockets:

17. Insert the 10k resistor network into the socket, note the “dot” indicating pin one:



18. Insert the CPU in U2, the RAM in U3 and the IO in U5 and the logic gate into U1



19. Attach the provided rubber feet to the bottom of the PCB.

20. Depending on the type of ROM used, configure J3 jumpers:
    a. 27C256 EPROM memory (blue jumpers on the left of J3) like this:

    b. AT28C256 EEPROM (jumpers should be placed to the right of J3).

    c. W27C512 EEPROM from Winbond: those as pretty cheap on eBay and are pin compatible with EPROM but can be erased electricaly (I'm using the cheap TL866 programmer). Jumpers should be configured as per example a. (EPROM) and since this memory is 64k, you should load the program into the programmer starting at address 0x8000h (as A15 is held high for the 27C256).

You "MGH80" system is ready to use, you can find a simple "blink" program in the GitHub repository.

Legal Notice: