Retroshield 8008 for Arduino Mega/Teensy

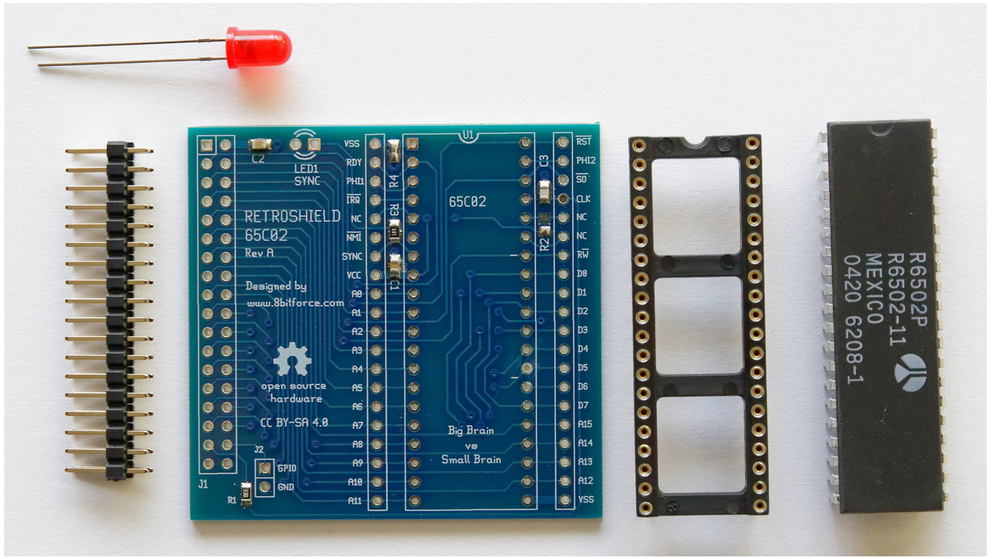
Breadboarding in software…

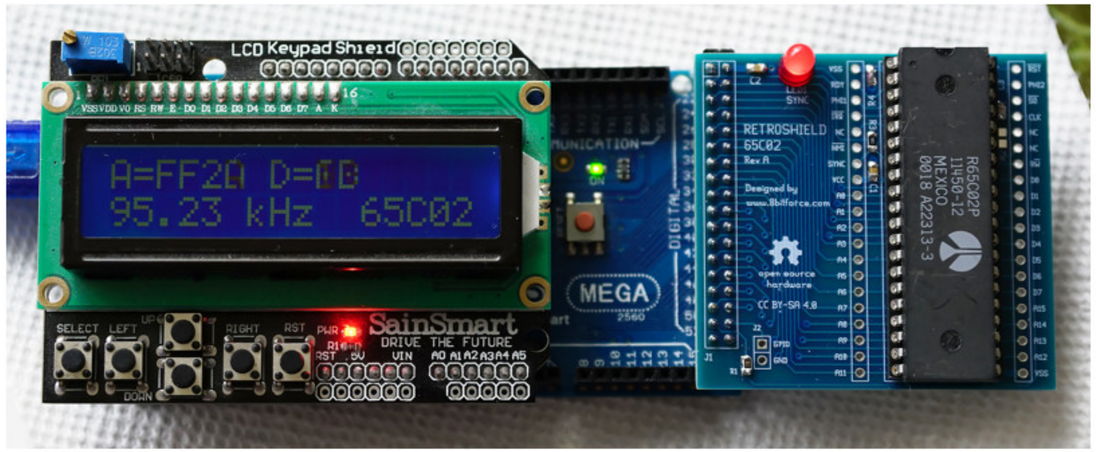
Erturk Kocalar
erturkk@8bitforce.com

https://gitlab.com/8bitforce

29 MAR 2025

Breadboarding in Software

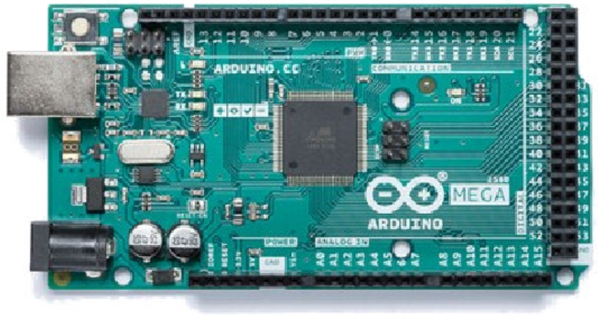


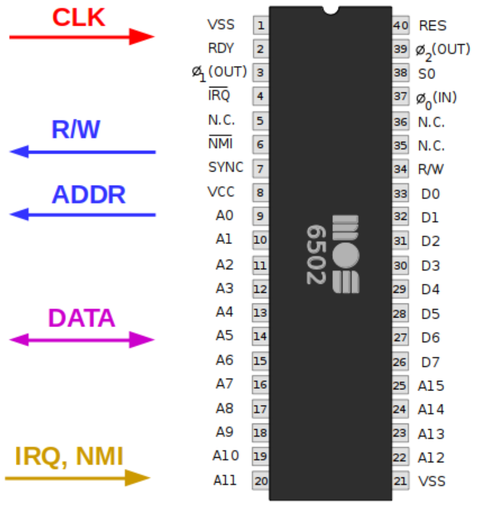


RetroShield for Arduino Mega/Teensy

Microprocessor daughter card that plugs into Arduino Mega/Teensy.

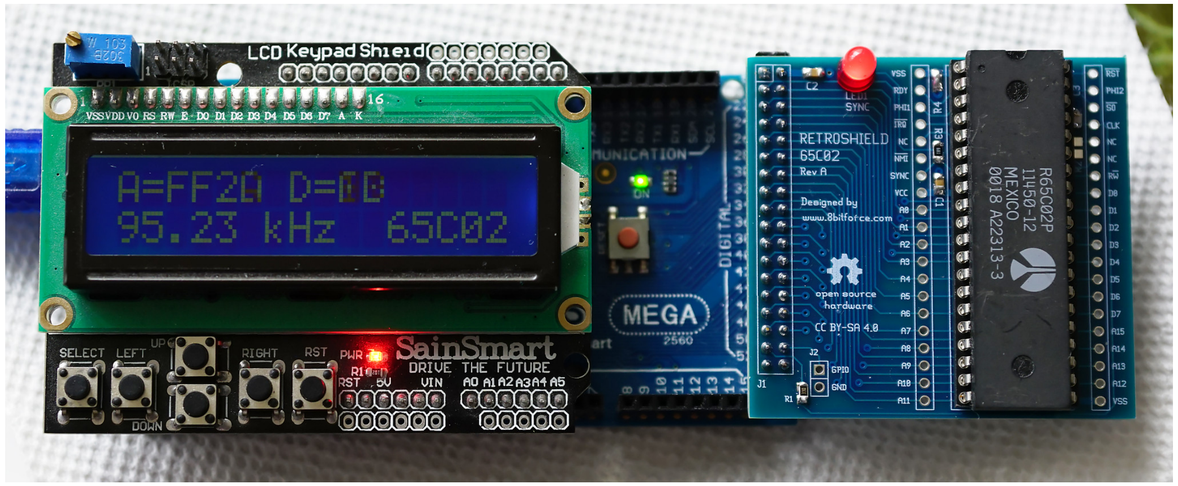
Real processor executes code while Arduino/Mega emulate system hw.
Switching system hardware is as easy as uploading new Arduino code.
Software development done in Arduino IDE, with Arduino shields & libraries.
Open source hardware/software.



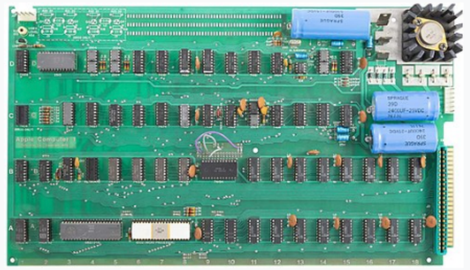


CPU Daughtercards already released:

- 1Bit: 14500B
- 4Bit: 4004, 4040
- 8Bit: 8008, 8031, 8080, 8085, 8088
- 8Bit: 6502, 6803, 6809
- 8Bit: 1802, 2650, Z80, INS8060(SC/MP II)
- 12Bit: HD6120 (PDP8)
- 16Bit: 68008

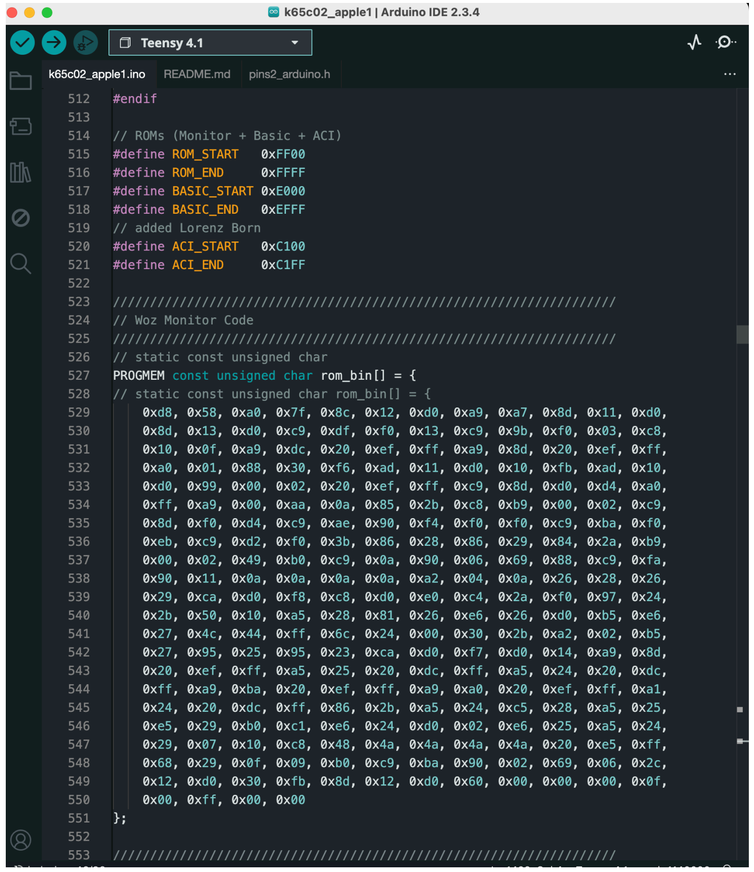


=>



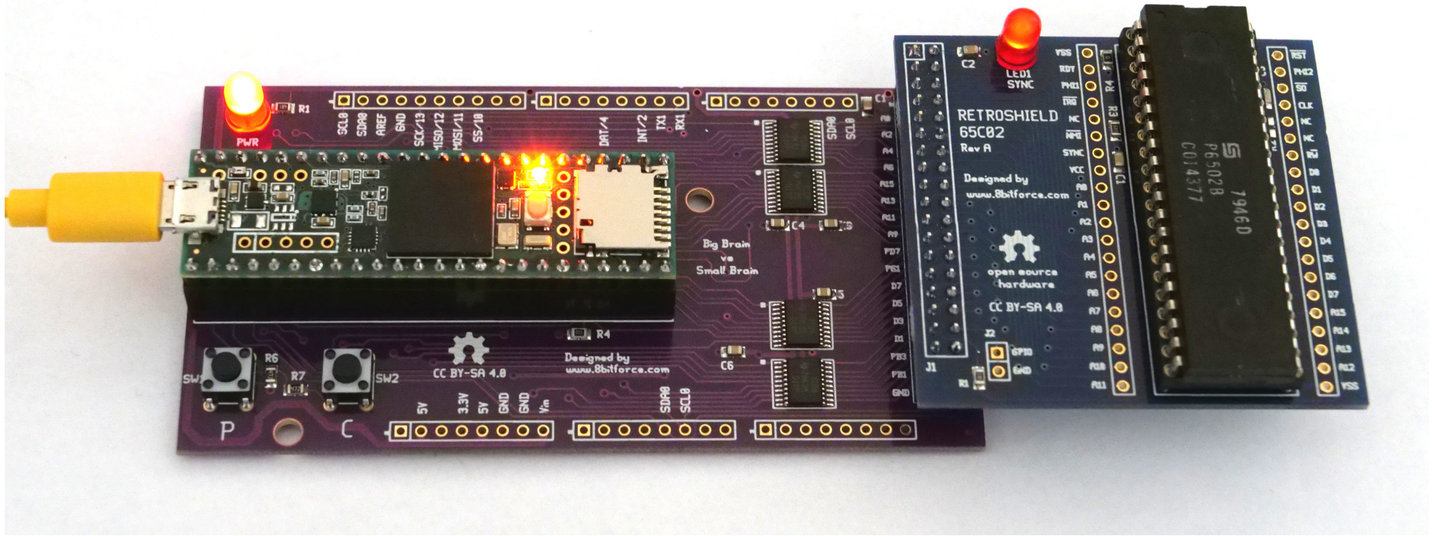
KIM-1

Credit Wikipedia



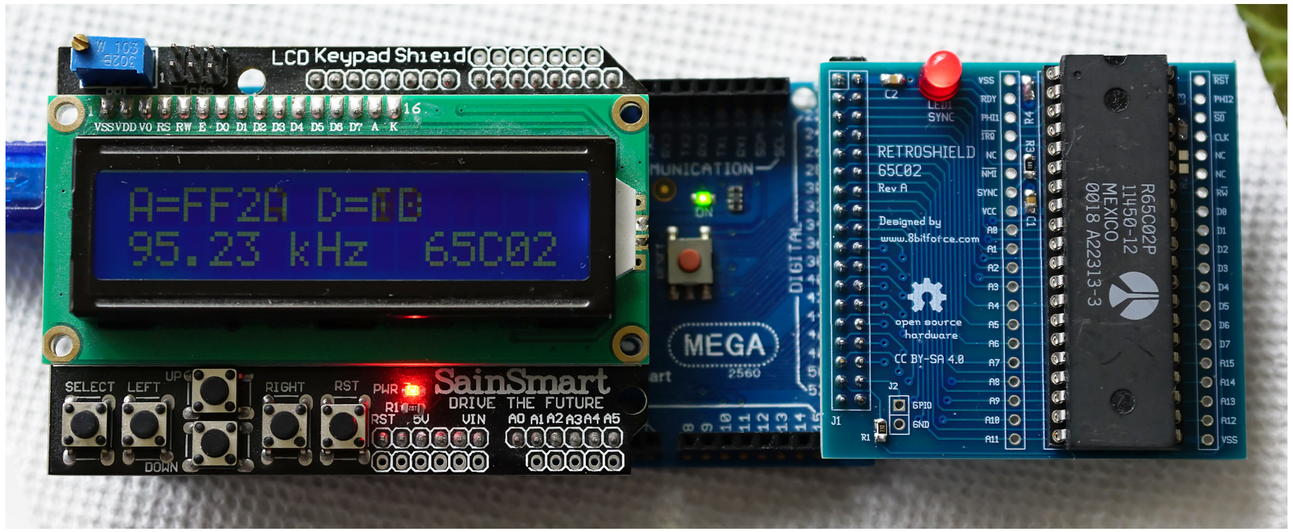
- **Experience** old computer systems in a cheap & easy way.
- Replace ROM contents easily by changing C code.
- Keep historical/valuable software alive.
- Design your own custom computer using software.
- Learn low level hardware interfacing w/ microprocessors.

Two platforms



Teensy 4.1 (600MHz) Fast Platform:

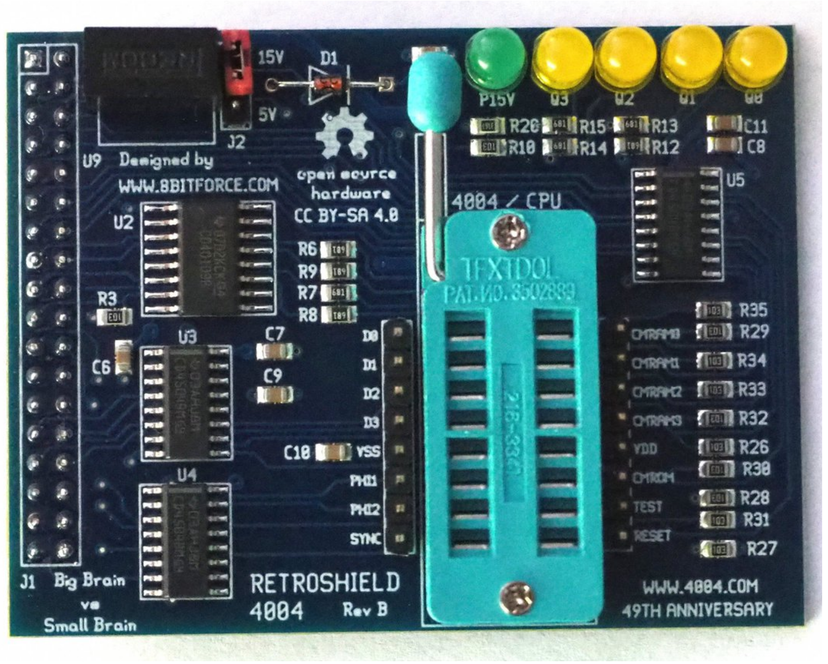
- Microprocessor can run at full speed, i.e. 1.00+ MHz
- **RAM : 1MB (Teensy 4.1), 256 KB (Teensy 3.5/3.6)**
- ROM : >512 KB
- **Onboard 3.3V<=>5V level shifting**
- microSD slot (can be used to emulate disk drives)
- Peripherals: UART, PIA, Timers, etc. (emulated by Teensy)

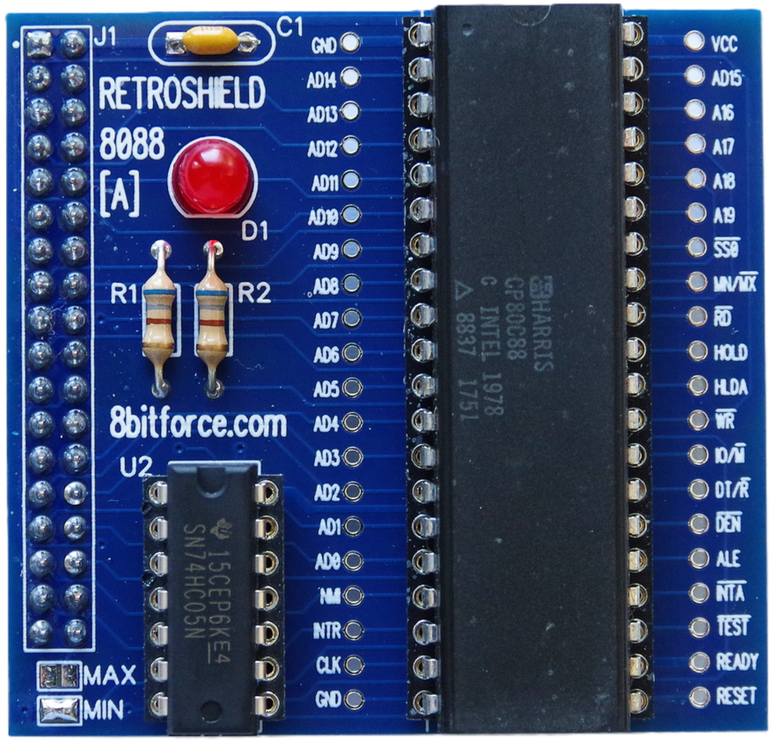


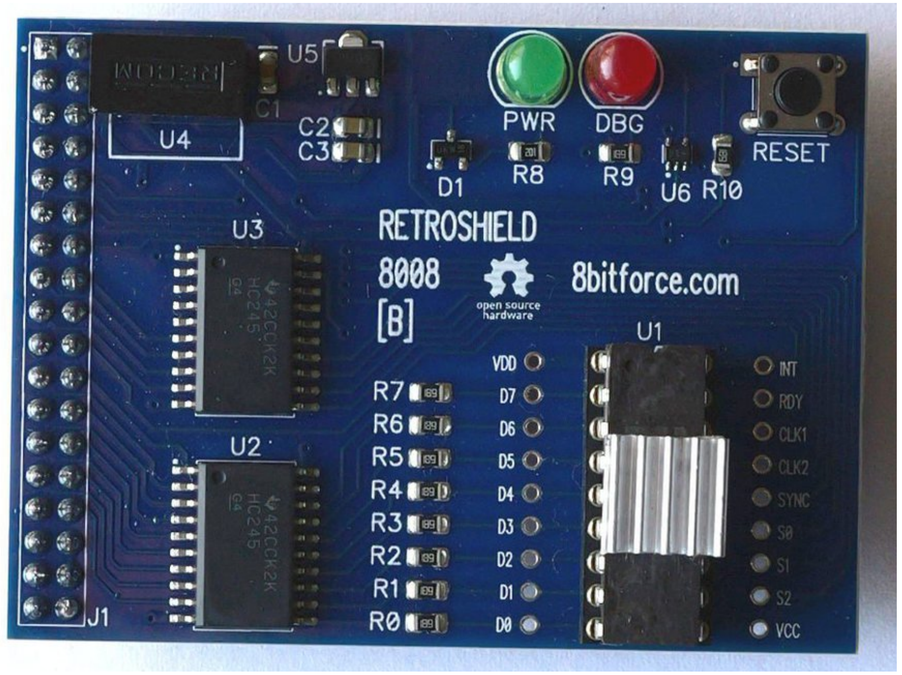
Arduino Mega (16MHz) Cheap Platform:

- Microprocessor runs at about 100~200kHz up to 400kHz.
- ROM : >200 KB
- **RAM : ~6 KB** (2KB of 8KB reserved for Arduino stack/heap)
- Peripherals: UART, PIA, Timers, etc. (emulated by Arduino)
- **Arduino Shields can be used to add new features.**

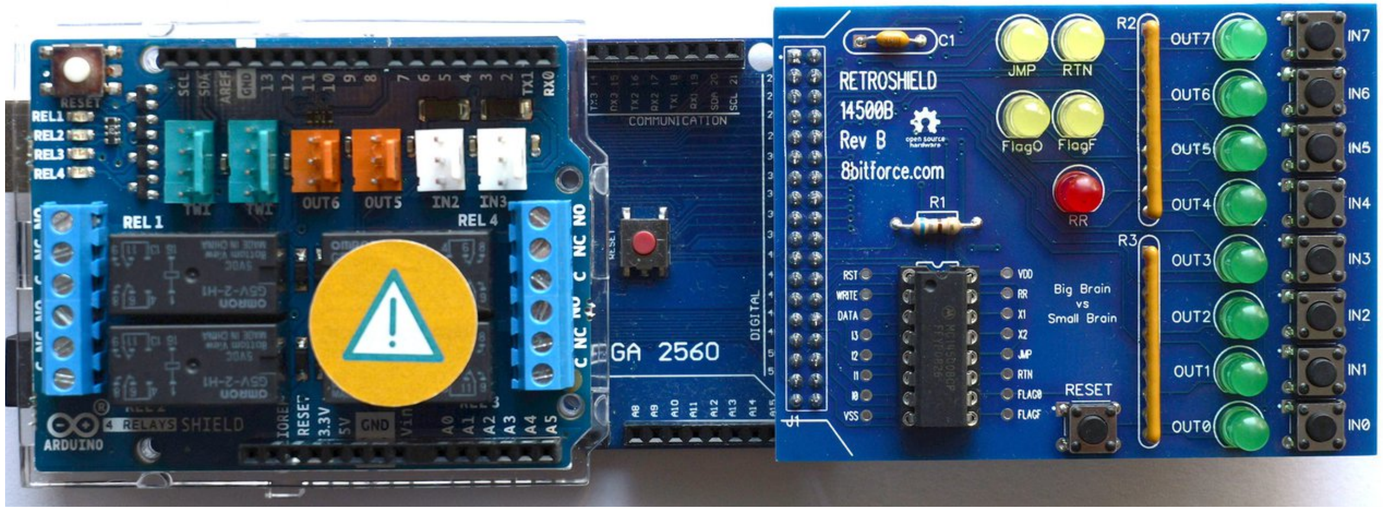
As of 2025-03-29: Mega 2560 clone $25. Teensy 4.1 ($31.50) + Teensy adapter ($24.00). Most 8bit cpu cards cost $24.00, some more expensive depending on processor.

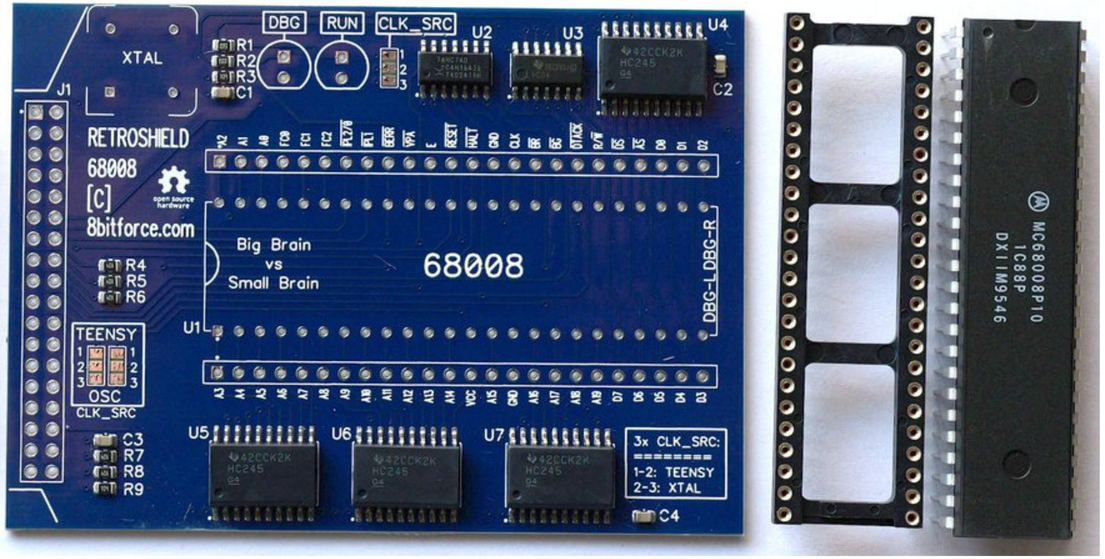






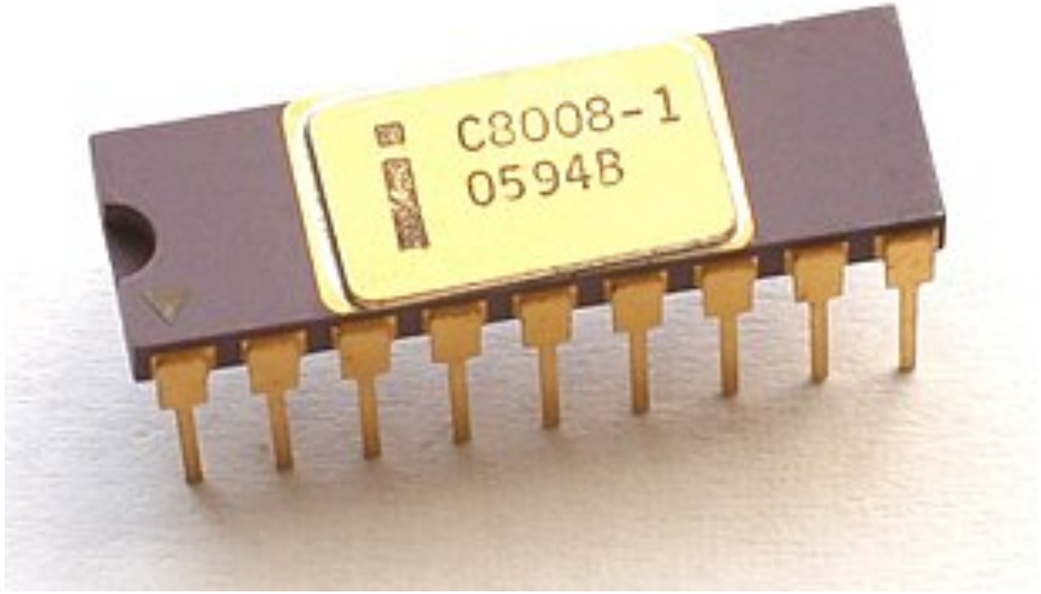
Retroshields come in many shapes and forms…





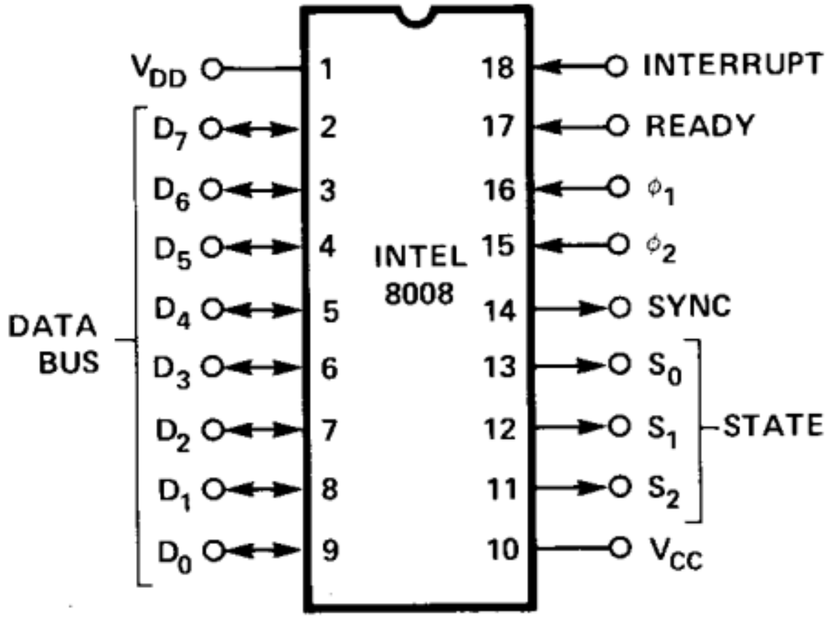
RetroShield 8008

World’s first 8-bit microprocessor



Credit Wikipedia

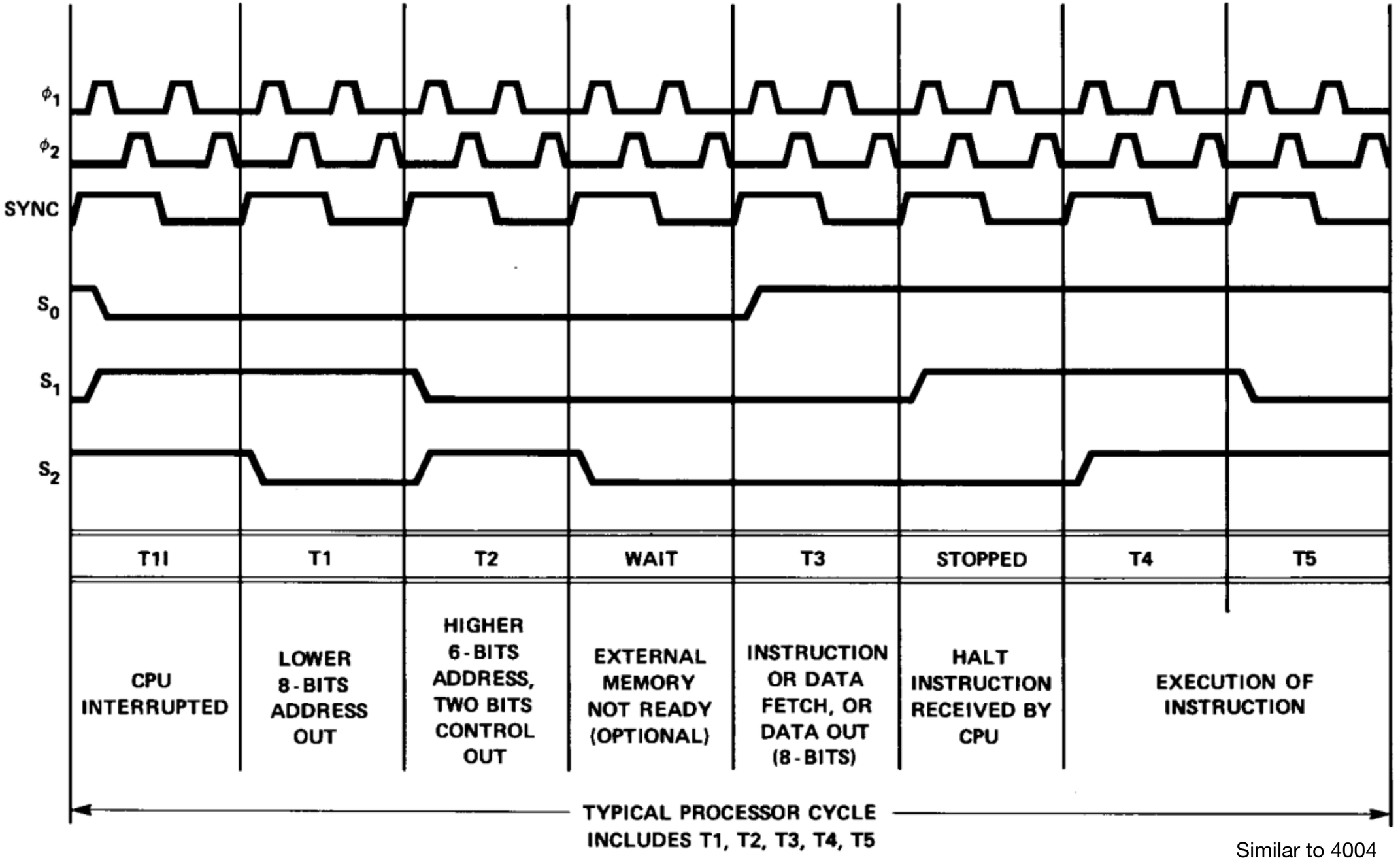
RetroShield 8008



- Two power supplies: +5V and -9V. No GND pin.
- INTERRUPT pin requires externally feeding one instruction for interrupt handler, usually RST.
- no RESET input. on power-up, interrupt pin used to feed the start-up instruction, i.e. RST (Restart) or JMP EXEC

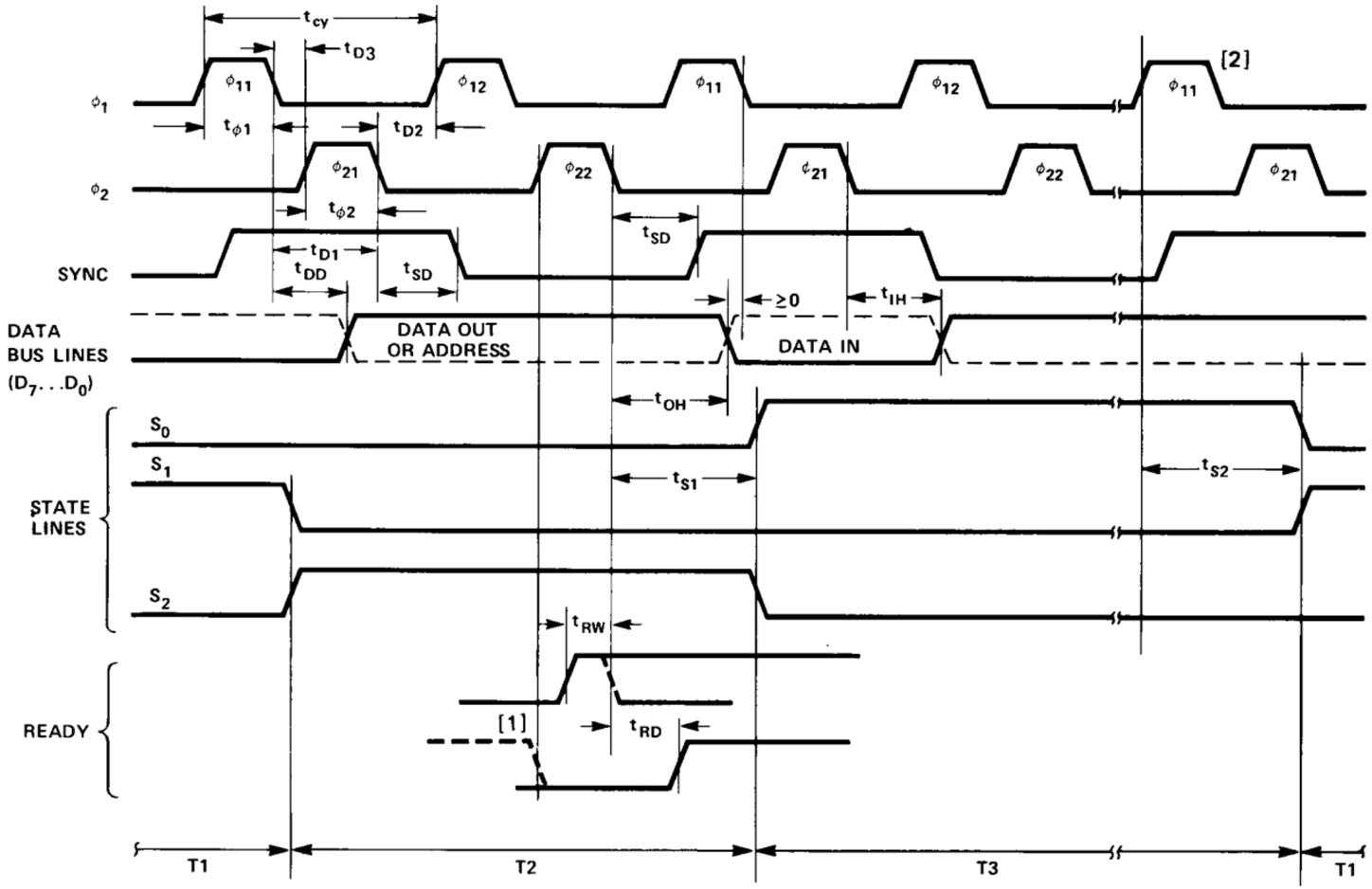
Intel 8008 registers

$^{1}_{3}$ $^{1}_{2}$ $^{1}_{1}$ $^{1}_{0}$ $^{0}_{9}$ $^{0}_{8}$ $^{0}_{7}$ $^{0}_{6}$ $^{0}_{5}$ $^{0}_{4}$ $^{0}_{3}$ $^{0}_{2}$ $^{0}_{1}$ $^{0}_{0}$ *(bit position)*		
Main registers		
	A	**A**ccumulator
	B	**B** register
	C	**C** register
	D	**D** register
	E	**E** register
	H	**H** register *(indirect)*
	L	**L** register *(indirect)*
Program counter		
PC		**P**rogram **C**ounter
Push-down address call stack		
AS		Call level 1
AS		Call level 2
AS		Call level 3
AS		Call level 4
AS		Call level 5
AS		Call level 6
AS		Call level 7
Flags		
C P Z S		Flags[12][a]



Similar to 4004

TIMING DIAGRAM



[1] READY line must be at "0" prior to ϕ_{22} of T_2 to guarantee entry into the WAIT state.

[2] INTERRUPT line must not change levels within 200ns (max.) of the falling edge of ϕ_1.

A.C. CHARACTERISTICS

T_A = 0°C to 70°C; V_{CC} = +5V ±5%, V_{DD} = –9V ±5%. All measurements are referenced to 1.5V levels.

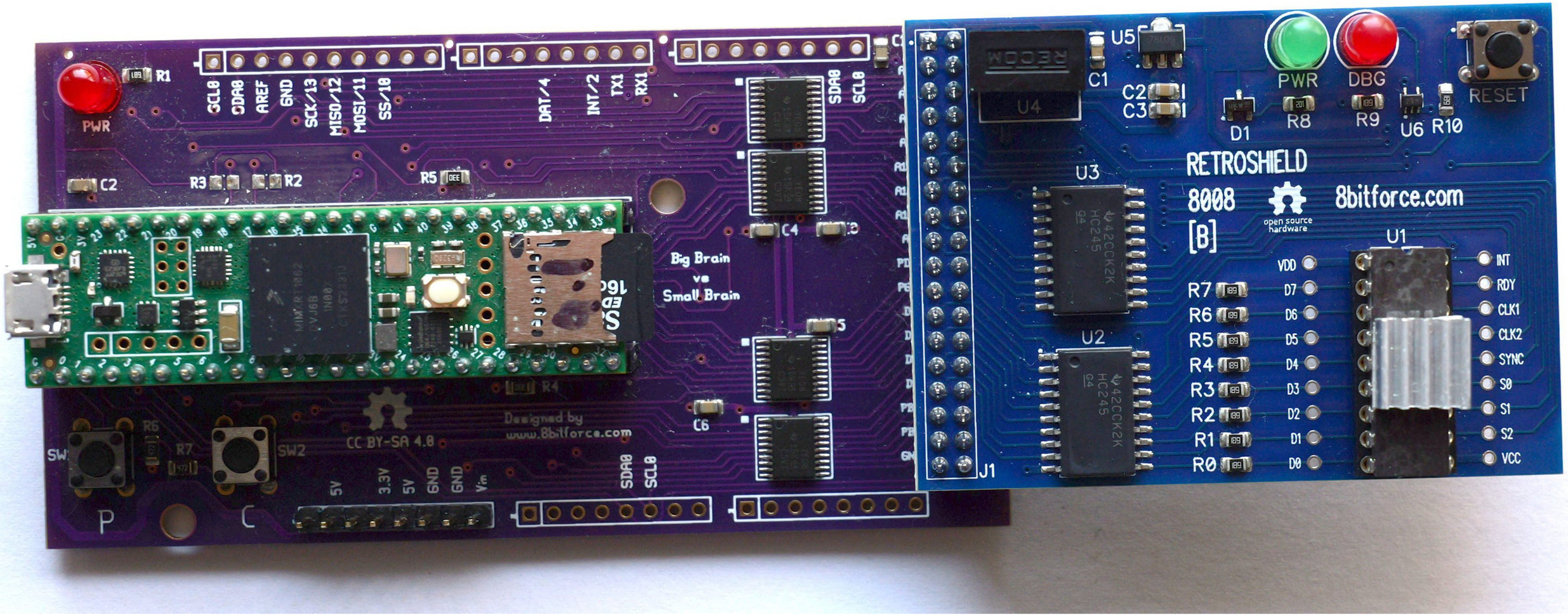
SYMBOL	PARAMETER	8008 LIMITS MIN.	8008 LIMITS MAX.	8008-1 LIMITS MIN.	8008-1 LIMITS MAX.	UNIT	TEST CONDITIONS
t_{CY}	CLOCK PERIOD	2	3	1.25	3	μs	t_R, t_F = 50ns
t_R, t_F	CLOCK RISE AND FALL TIMES		50		50	ns	
$t_{\phi 1}$	PULSE WIDTH OF ϕ_1	.70		.35		μs	
$t_{\phi 2}$	PULSE WIDTH OF ϕ_2	.55		.35		μs	
t_{D1}	CLOCK DELAY FROM FALLING EDGE OF ϕ_1 TO FALLING EDGE OF ϕ_2	.90	1.1		1.1	μs	
t_{D2}	CLOCK DELAY FROM ϕ_2 TO ϕ_1	.40		.35		μs	
t_{D3}	CLOCK DELAY FROM ϕ_1 TO ϕ_2	.20		.20		μs	
t_{DD}	DATA OUT DELAY		1.0		1.0	μs	C_L = 100pF
t_{OH}	HOLD TIME FOR DATA OUT	.10		.10		μs	
t_{IH}	HOLD TIME FOR DATA IN	[1]		[1]		μs	
t_{SD}	SYNC OUT DELAY		.70		.70	μs	C_L = 100pF
t_{S1}	STATE OUT DELAY (ALL STATES EXCEPT T1 AND T1I) [2]		1.1		1.1	μs	C_L = 100pF
t_{S2}	STATE OUT DELAY (STATES T1 AND T1I)		1.0		1.0	μs	C_L = 100pF
t_{RW}	PULSE WIDTH OF READY DURING ϕ_{22} TO ENTER T3 STATE	.35		.35		μs	
t_{RD}	READY DELAY TO ENTER WAIT STATE	.20		.20		μs	

[1] t_{IH} MIN ≥ t_{SD} [2] If the INTERRUPT is not used, all states have the same output delay, t_{S1}.

Pay attention - clock must be between 333kHz ... 800kHz.

Reset button goes to Teensy to trigger interrupt/reset.

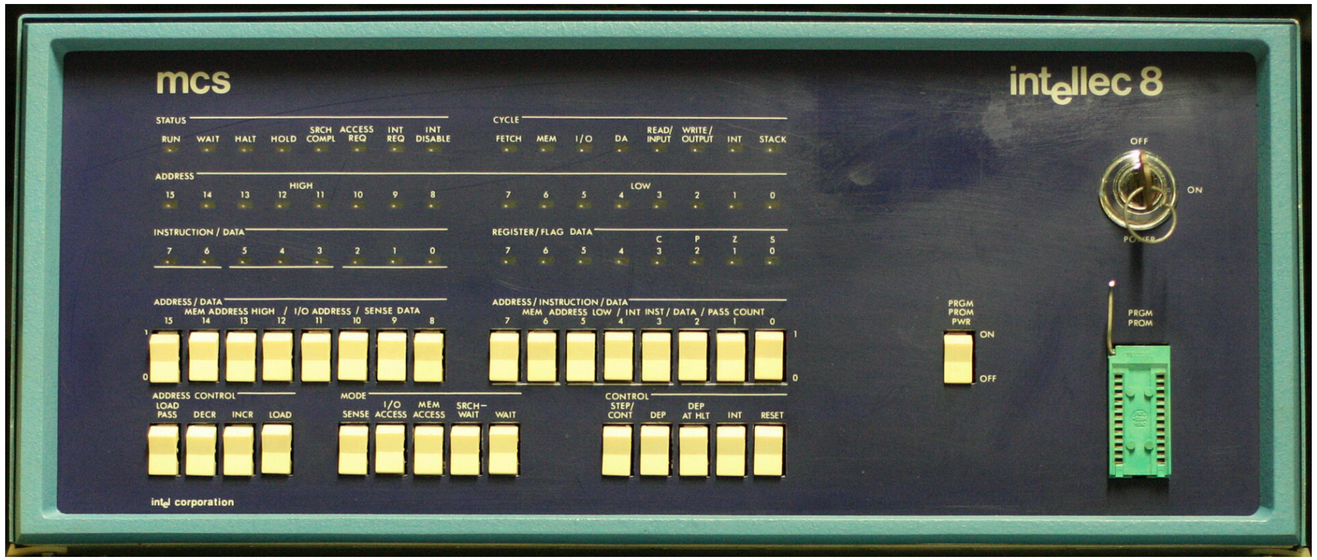
-9V Supply



Level Shifters

8008+Heatsink

8008 Uses



INTELLEC 8/MOD 8

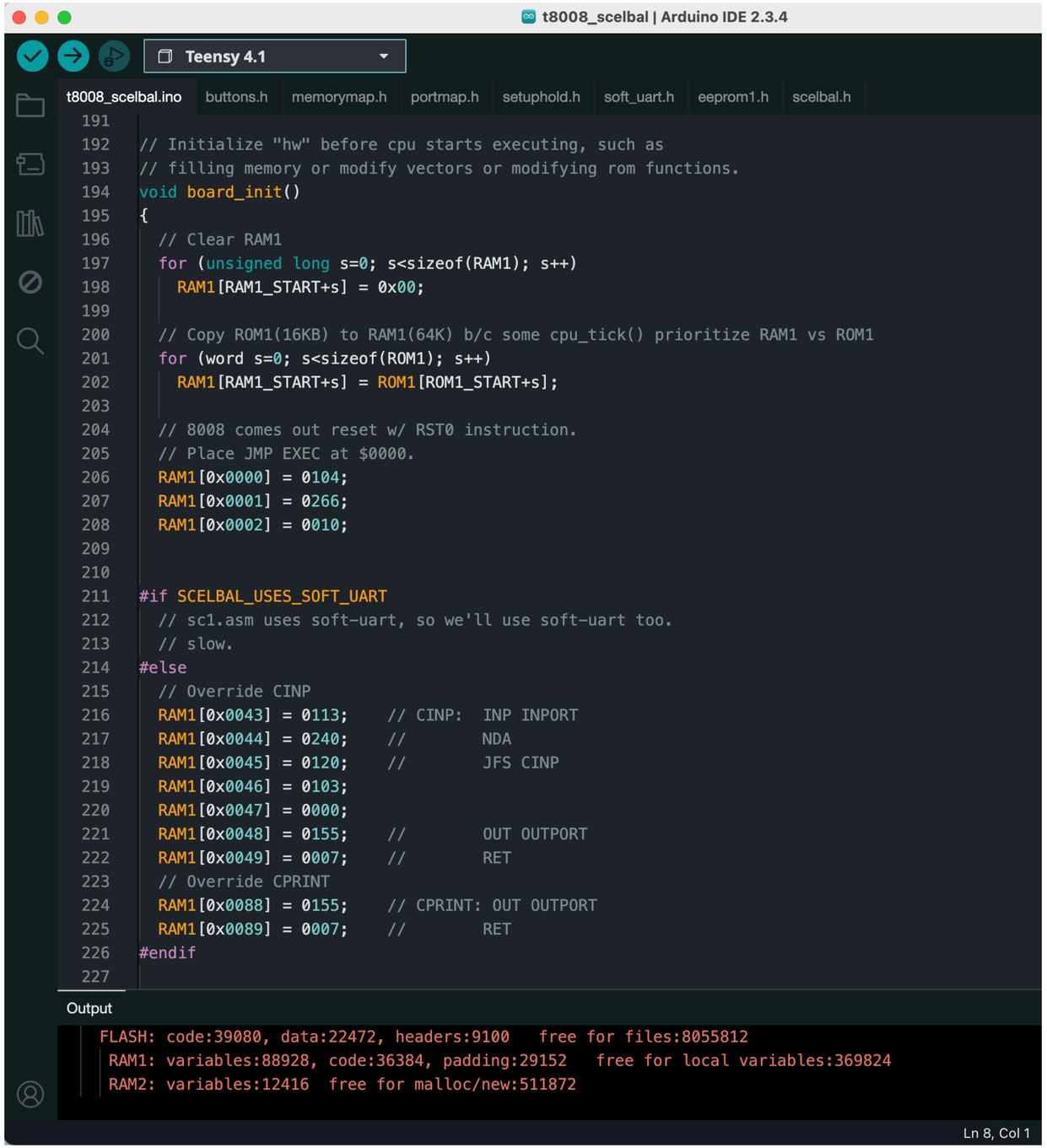
- ROM-Resident monitor (memory, tape/punch, tty)
- Editor/Assembler available on tape (?lost)
- PROM programmer

Credit Wikipedia

SCELBI

- SCELBAL (SCientific ELementary BAsic Language)
- Book form w/ very well documented assembly
- Floating point support
- Math functions (Sin, Cos, Exp, Log, Atn)
- String and array DIM

Credit https://www.scelbi.com/



board_init() prepares RAM before 8008 starts:

0) Copies EEPROM to RAM

1) Add JMP EXEC instruction at $0000.

2) Override UART functions for speed.

3) Preload BASIC program in memory.

```
229   // Save a default program in memory.
230   // each line entry has i) length of line, ii) line it self.
231 #if (1)
232   word prog_addr = 0x1b00;
233   prog_addr = save_basic_line(prog_addr, "10 PRINT 'RADIUS';");
234   prog_addr = save_basic_line(prog_addr, "15 INPUT R");
235   prog_addr = save_basic_line(prog_addr, "16 K=1.6");
236   prog_addr = save_basic_line(prog_addr, "20 R2=R^2");
237   prog_addr = save_basic_line(prog_addr, "21 PRINT 'AREA =';3.14159*R2;' CIRCUM =';3.14159*2*R");
238   prog_addr = save_basic_line(prog_addr, "22 PRINT");
239   prog_addr = save_basic_line(prog_addr, "23 PRINT");
240   prog_addr = save_basic_line(prog_addr, "25 K1=K*R");
241   prog_addr = save_basic_line(prog_addr, "30 FOR X=-R TO R+0.1");
242   prog_addr = save_basic_line(prog_addr, "40 Y=K*SQR(R2-X^2)");
243   prog_addr = save_basic_line(prog_addr, "50 PRINT TAB(2.5+K1-Y);'*';TAB(5.5+K1+Y);'*'");
244   prog_addr = save_basic_line(prog_addr, "60 NEXT X");
245   prog_addr = save_basic_line(prog_addr, "70 END");
246 #endif
247
```

SCELBAL

Untitled_0

New Open Save Connect Disconnect Options Clear Data View Help

```
Teensy:     4.1
Debug:      0
--------------
--------------
SRAM1 Size:  65536 Bytes
SRAM1_START: 0x0
SRAM1_END:   0xFFFF
--------------

=====================================================
; SCELBAL by Mark G. Arnold (MGA) and Nat Wadsworth
; Copyright 1975 Scelbi Computer Consulting, Inc.
; All rights reserved
;
; SCELBAL is the only open-source, floating-point
; high-level language ever implemented on Intel's first
; general-purpose microprocessor, the 8008.  It was.
; published in book form:
; SCELBAL: A Higher-Level Language for 8008/8080 Systems
;
; ...This code originates from a version made by
;    Steve Loboyko in 2001.
;    This version has all 3 patches for SCELBAL.
;
; Big thanks to MGA for Retroshield use. 2025/02/24.
=====================================================

=> type LIST to see the example drawing program.
.
LIST - see example circle program
RUN  - run the example circle program
SCR  - SCRatch. Start new program.
RESET Sequence - Initiate.
.
READY
```

Already loaded into memory.

```
LIST

10 PRINT 'RADIUS';
15 INPUT R
16 K=1.6
20 R2=R^2
21 PRINT 'AREA =';3.14159*R2;' CIRCUM =';3.14159*2*R
22 PRINT
23 PRINT
25 K1=K*R
30 FOR X=-R TO R+0.1
40 Y=K*SQR(R2-X^2)
50 PRINT TAB(2.5+K1-Y);'*';TAB(5.5+K1+Y);'*'
60 NEXT X
70 END
.
READY
```

usbmodem156369501 (Teensyduino) / 115200 8-N-1
Connected 00:00:19, 1,687 / 5 bytes

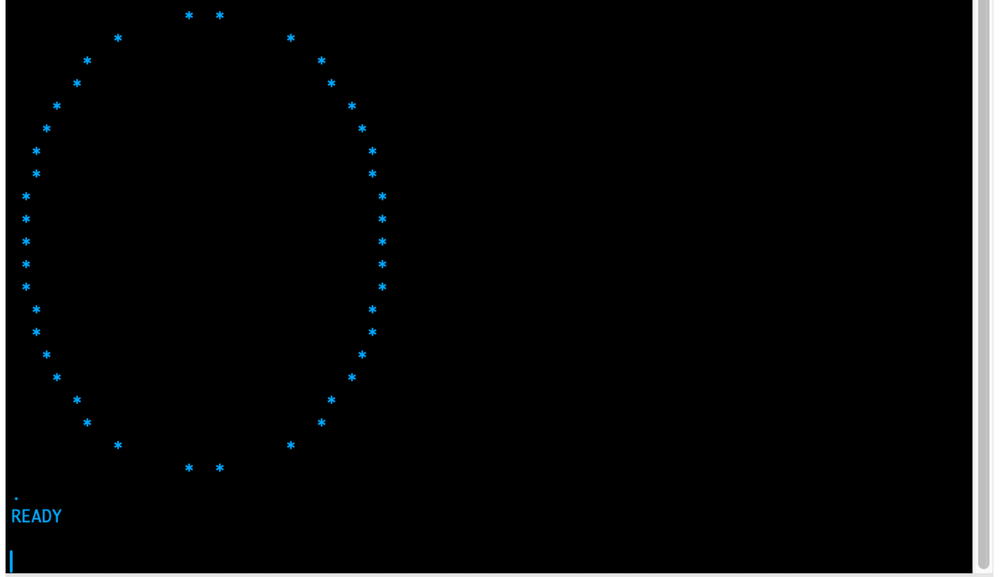
TX RX RTS CTS DTR DSR

App St

Untitled_0

New Open Save Connect Disconnect Options Clear Data View Help

```
SCR  - SCRatch. Start new program.
RESET Sequence - Initiate.
.
READY

LIST

10 PRINT 'RADIUS';
15 INPUT R
16 K=1.6
20 R2=R^2
21 PRINT 'AREA =';3.14159*R2;' CIRCUM =';3.14159*2*R
22 PRINT
23 PRINT
25 K1=K*R
30 FOR X=-R TO R+0.1
40 Y=K*SQR(R2-X^2)
50 PRINT TAB(2.5+K1-Y);'*';TAB(5.5+K1+Y);'*'
60 NEXT X
70 END
.
READY

RUN

RADIUS?10

AREA = 314.1592 CIRCUM = 62.83185
```

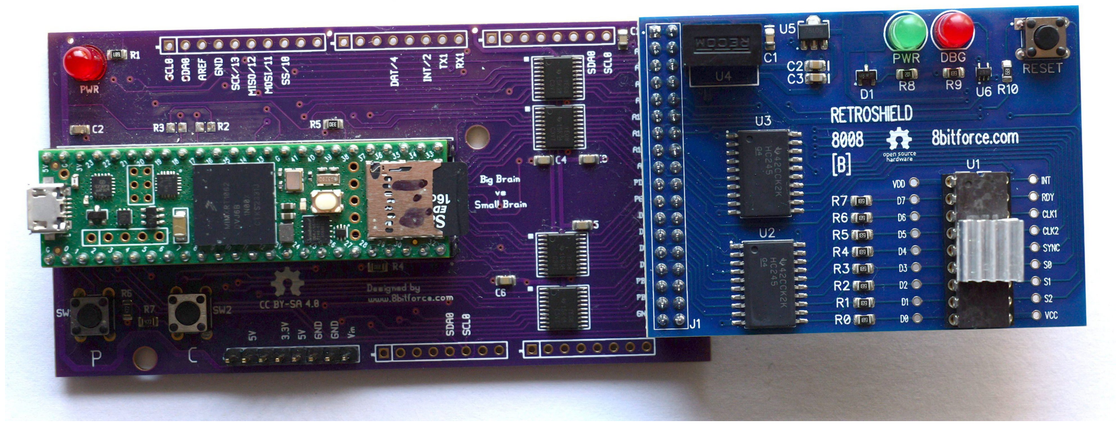


```
.
READY
```

usbmodem156369501 (Teensyduino) / 115200 8-N-1
Connected 00:01:28, 2,488 / 12 bytes

TX RX RTS CTS DTR DSR

Ap



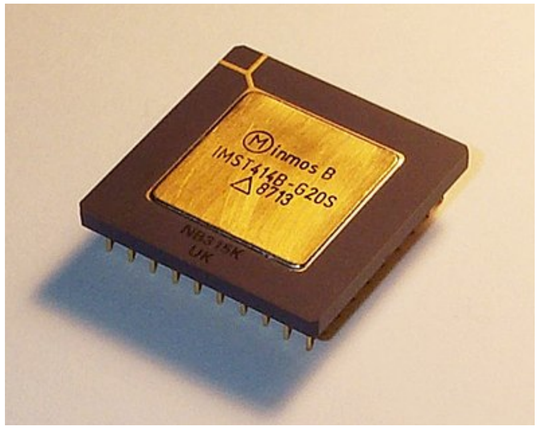
- Play w/ a wide range of processors and software from 50yrs ago.
  - Design your "hardware" and hack it till cows come home.
    - Invent your own "peripherals" with zero cost $ (DMA, UART, I/O)
      - Play w/ cpu bus interface timings to see where cpu breaks.
        - Print low level bus activity cycle by cycle.
          - Be creative and squeeze out every clock for processor speed.

Special Thanks to Mark G. Arnold (MGA)

- We have a rich computer history. As time goes on, naturally, companies shutdown, people leave, devices and files get fewer and even get lost.
- We can not steal their hard work just because they are not around.
- Companies and authors open-sourcing their commercial products is valuable. It enables new generation to experience and appreciate their products.
- I want to give special thanks to Mark G. Arnold (MGA) and the late Nat Wadsworth for enabling me to port SCELBAL to Retroshield 8008.
- I'm sure they weren't expecting SCELBAL to "gain" marketshare after 50 yrs.

What's next for me?

- Every year I tell myself I will be done…
- 2025: Working on next batch of processors
  - **Transputer** T805, T425, T225.
  - Intel **8080** (to complete the Intel series)
  - Texas Instruments **TMS9995**
  - **Holding pattern**: NS32016, 8X305, DEC T-11, Fairchild F8, Motorola 6800



Credit Wikipedia