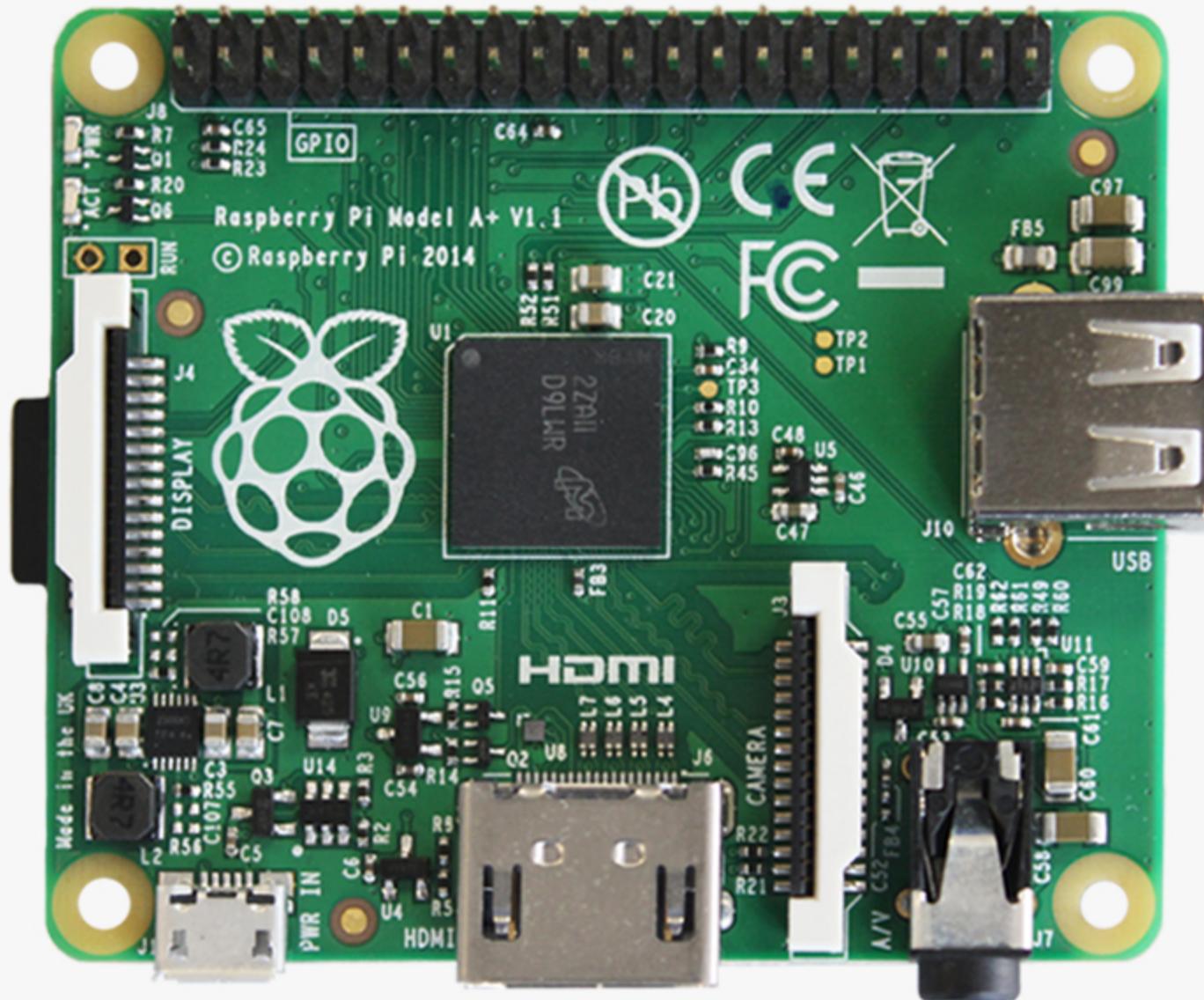




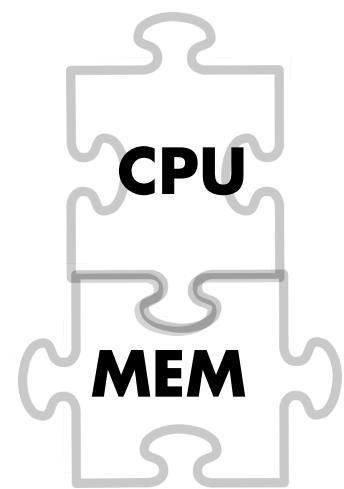
Goal 1

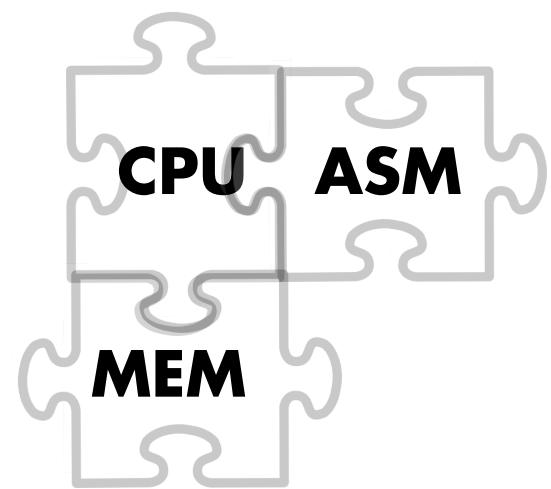
Understand How Computers Work

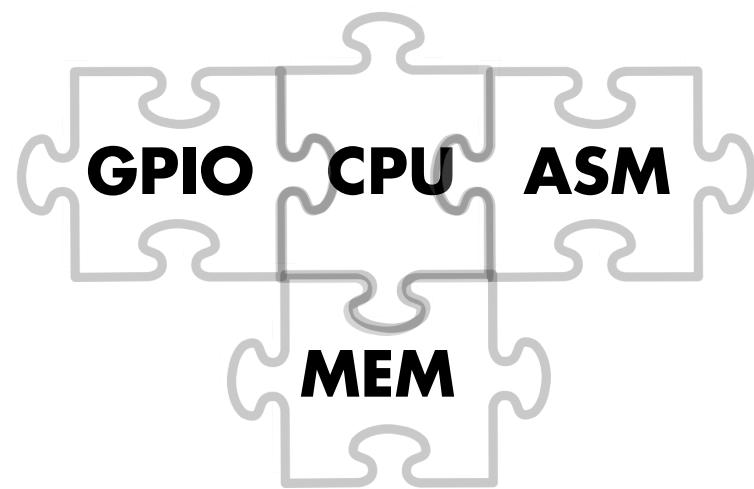
Computer Systems from the Ground Up

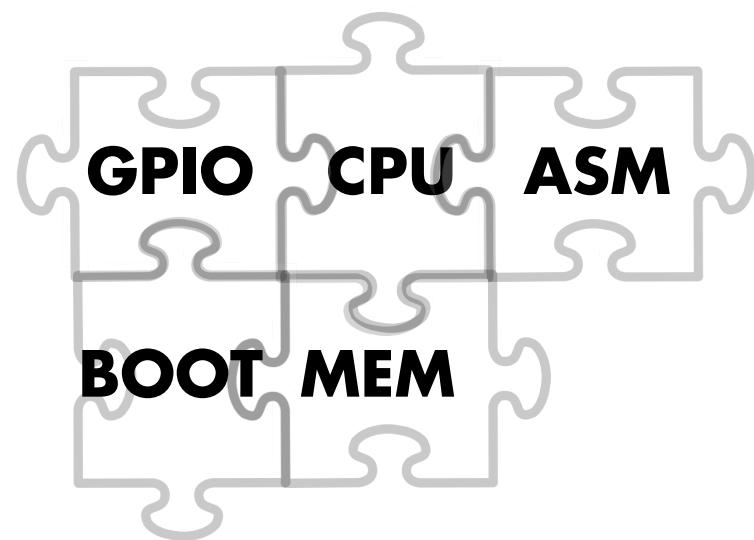


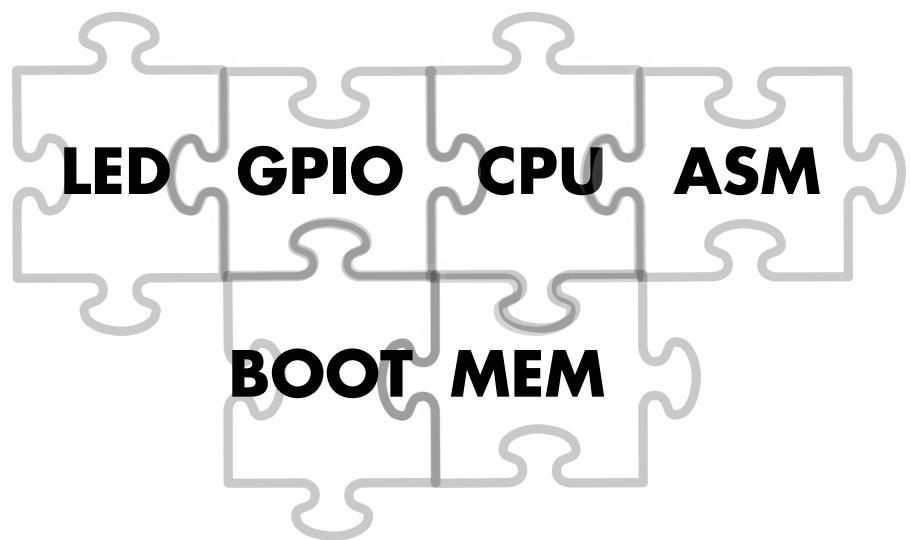
Bare Metal on the Raspberry Pi

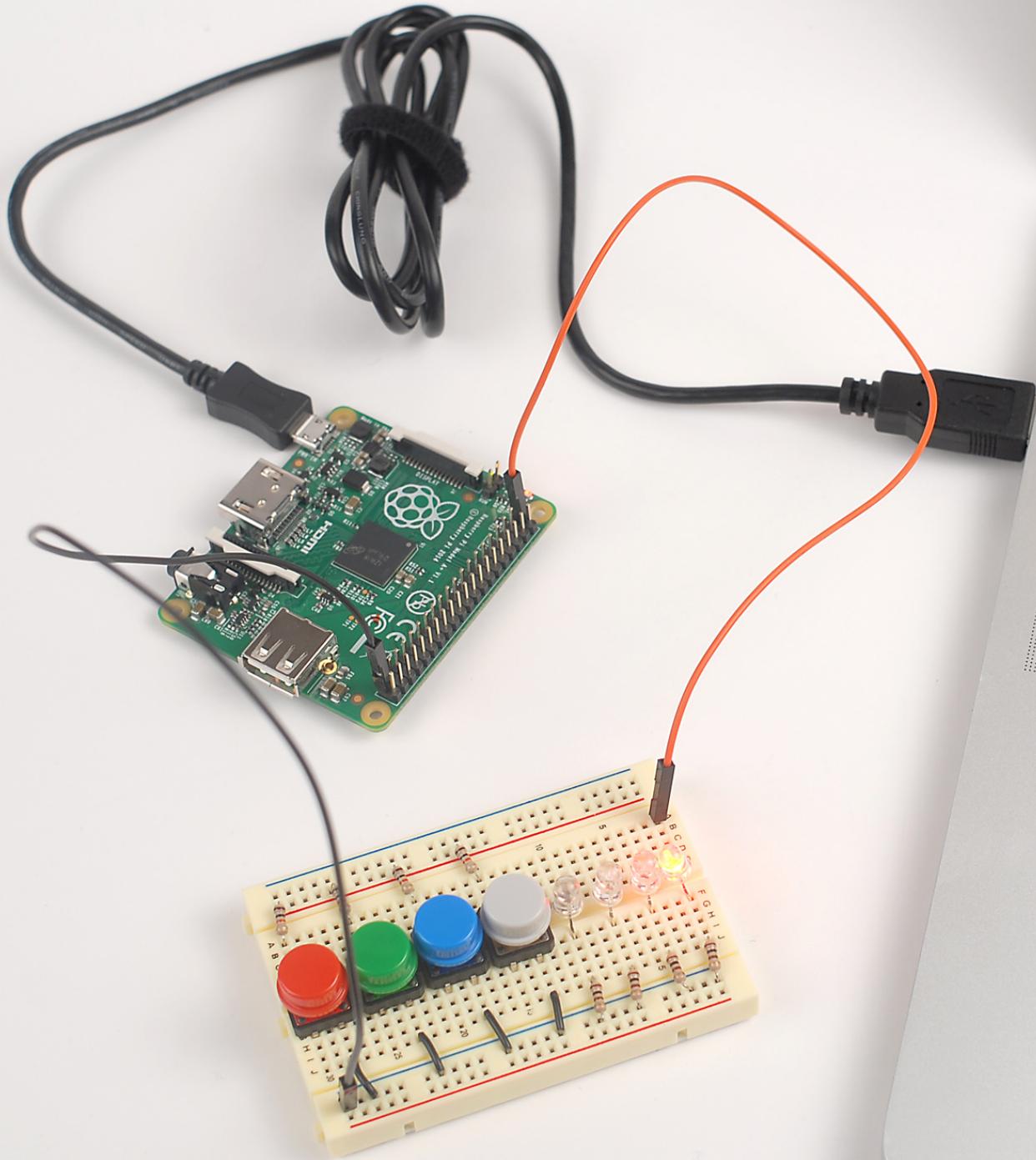


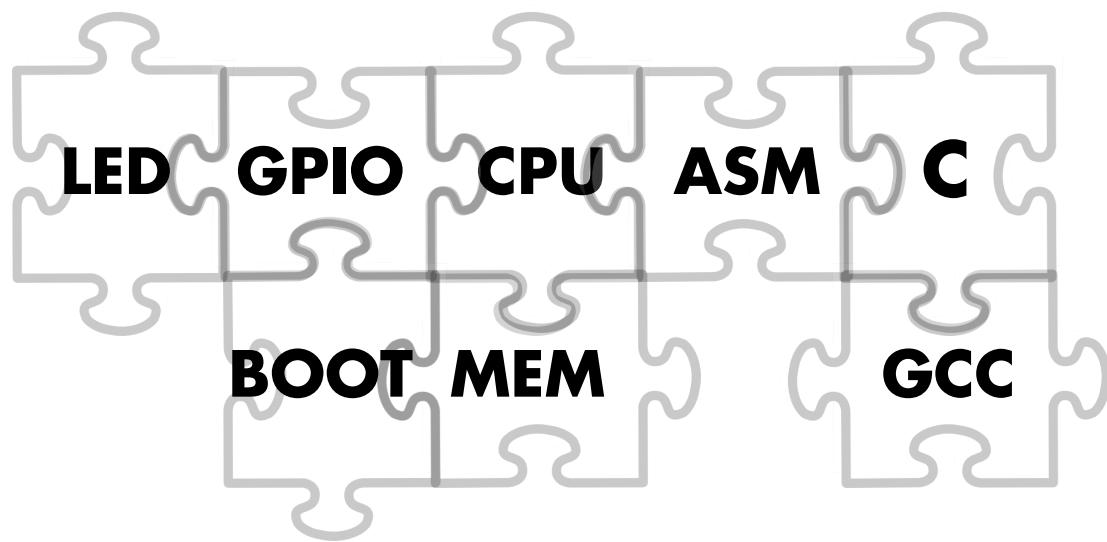


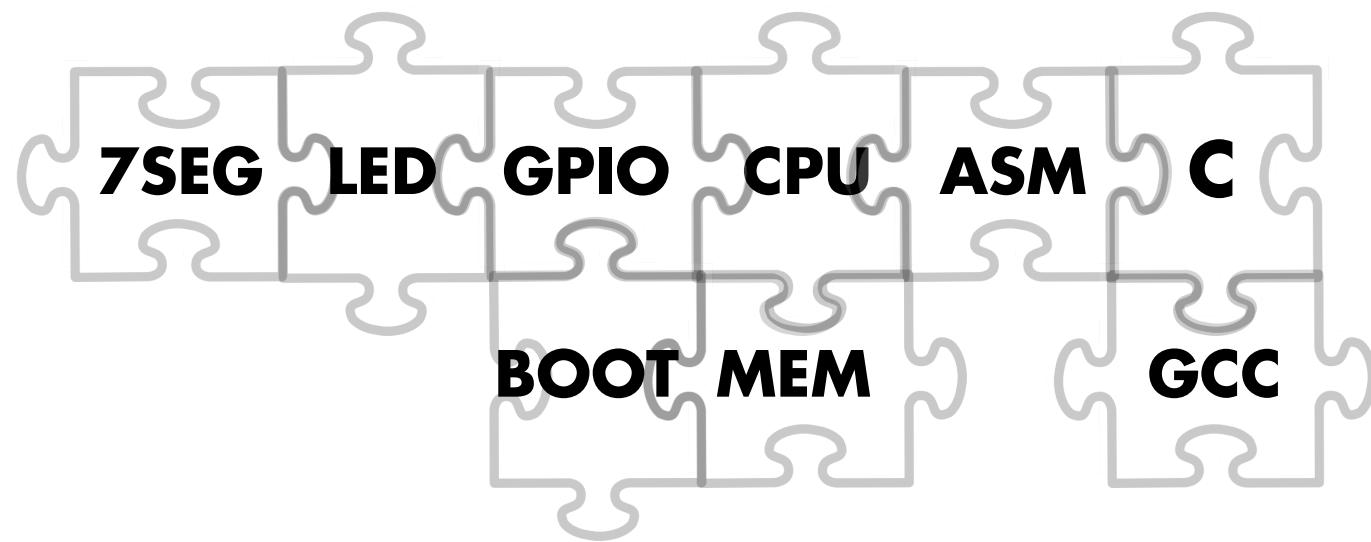


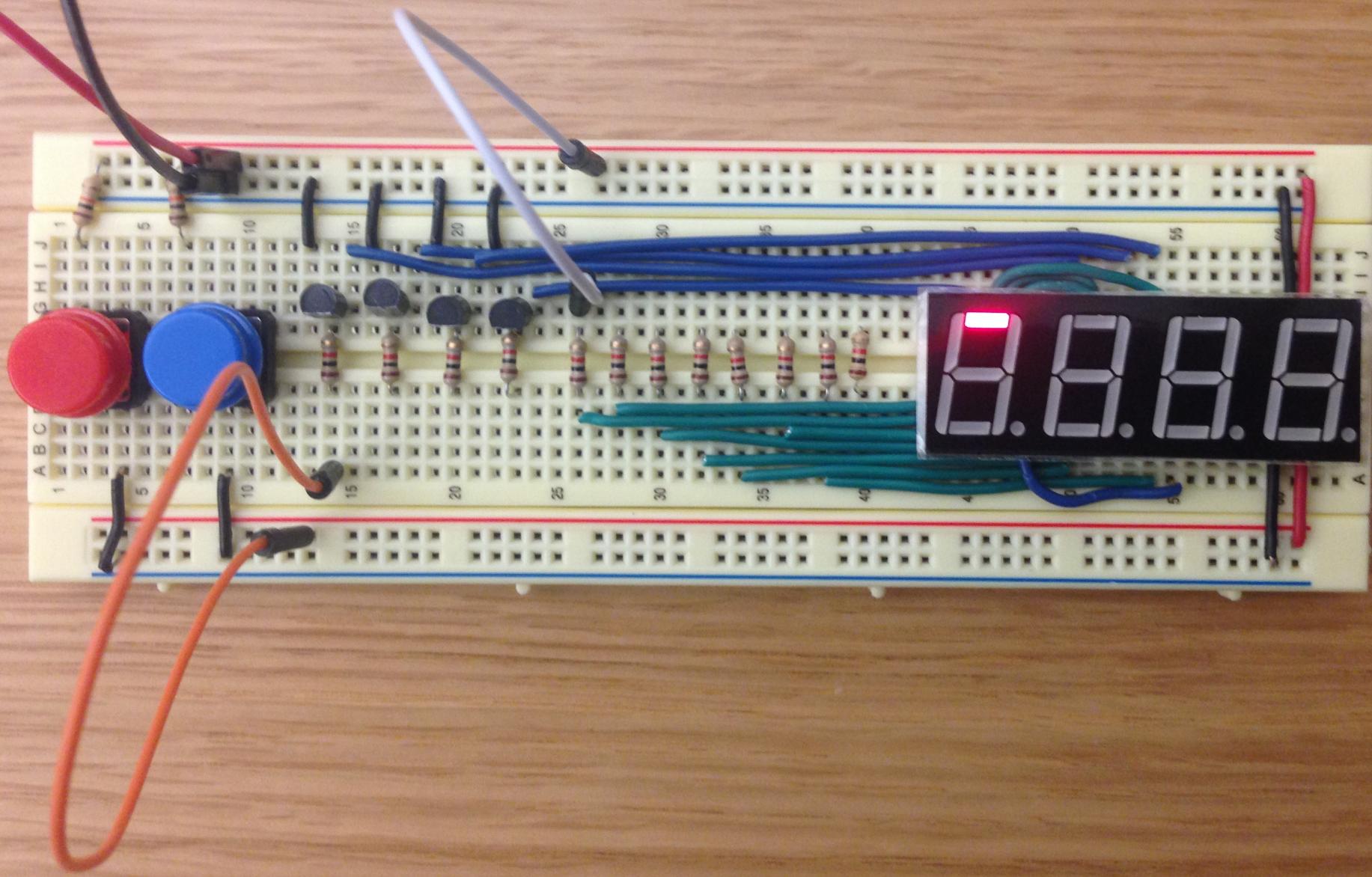












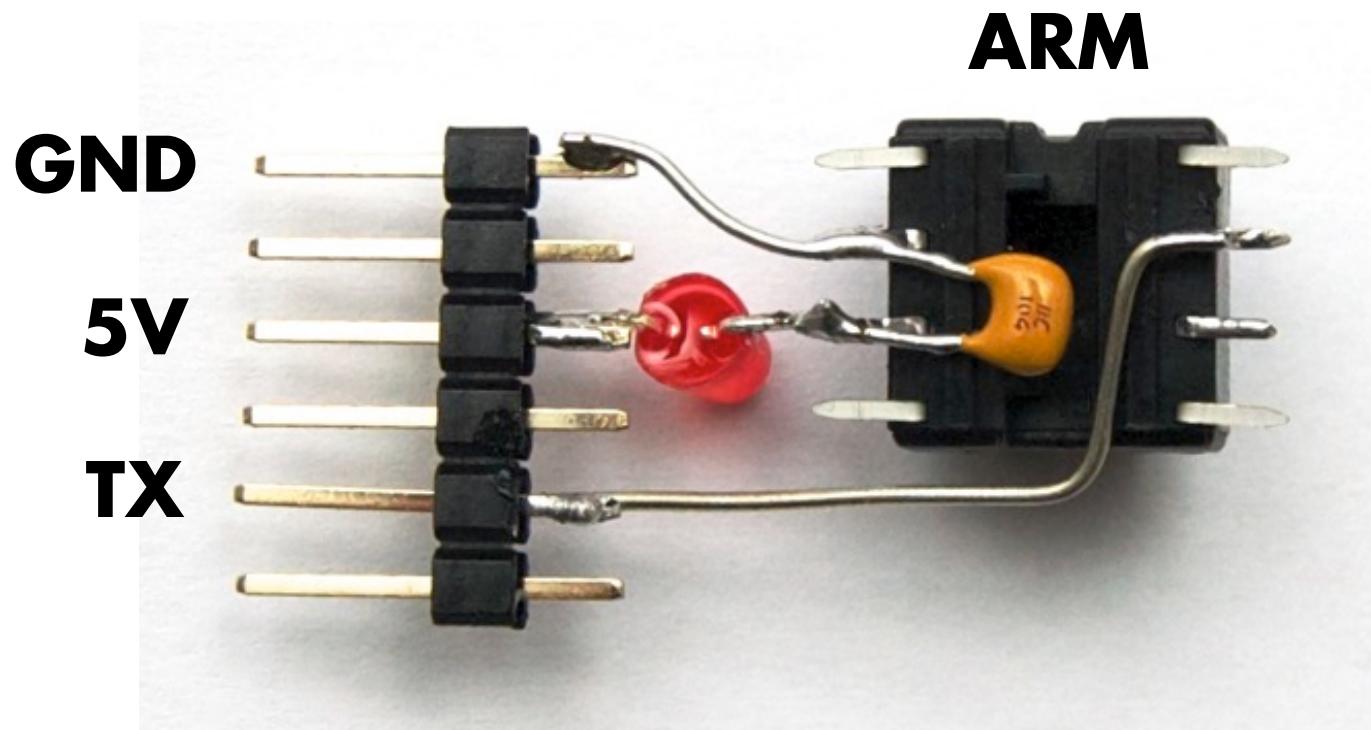
Simple and Clear

Functionality

is

Beautiful

Jean-Claude Wippler



www.jeelabs.org

```

/* match: search for regexp anywhere in text */
int match(char *regexp, char *text) {
    if (regexp[0] == '^')
        return matchhere(regexp+1, text);
    do { /* must look even if string is empty */
        if (matchhere(regexp, text))
            return 1;
    } while (*text++ != '\0');
    return 0;
}
/* matchhere: search for regexp at beginning of text */
int matchhere(char *regexp, char *text) {
    if (regexp[0] == '\0')
        return 1;
    if (regexp[1] == '*')
        return matchstar(regexp[0], regexp+2, text);
    if (regexp[0] == '$' && regexp[1] == '\0')
        return *text == '\0';
    if (*text != '\0' && (regexp[0] == '.' || regexp[0] == *text))
        return matchhere(regexp+1, text+1);
    return 0;
}
/* matchstar: search for c*regexp at beginning of text*/
int matchstar(int c, char *regexp, char *text) {
    do { /* a * matches zero or more instances */
        if (matchhere(regexp, text))
            return 1;
    } while (*text != '\0' && (*text++ == c || c == '.' ));
    return 0;
}

```

Ken Thompson

Regular Expression

ch

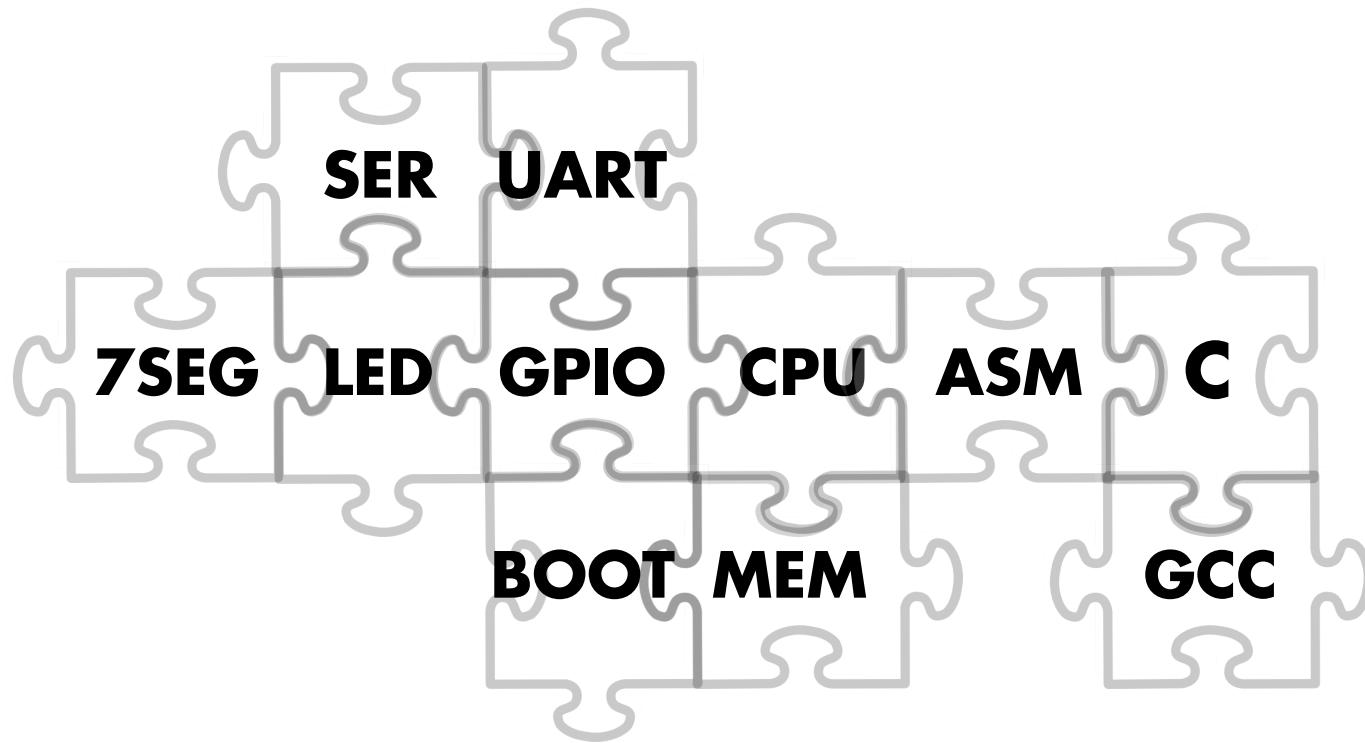
.

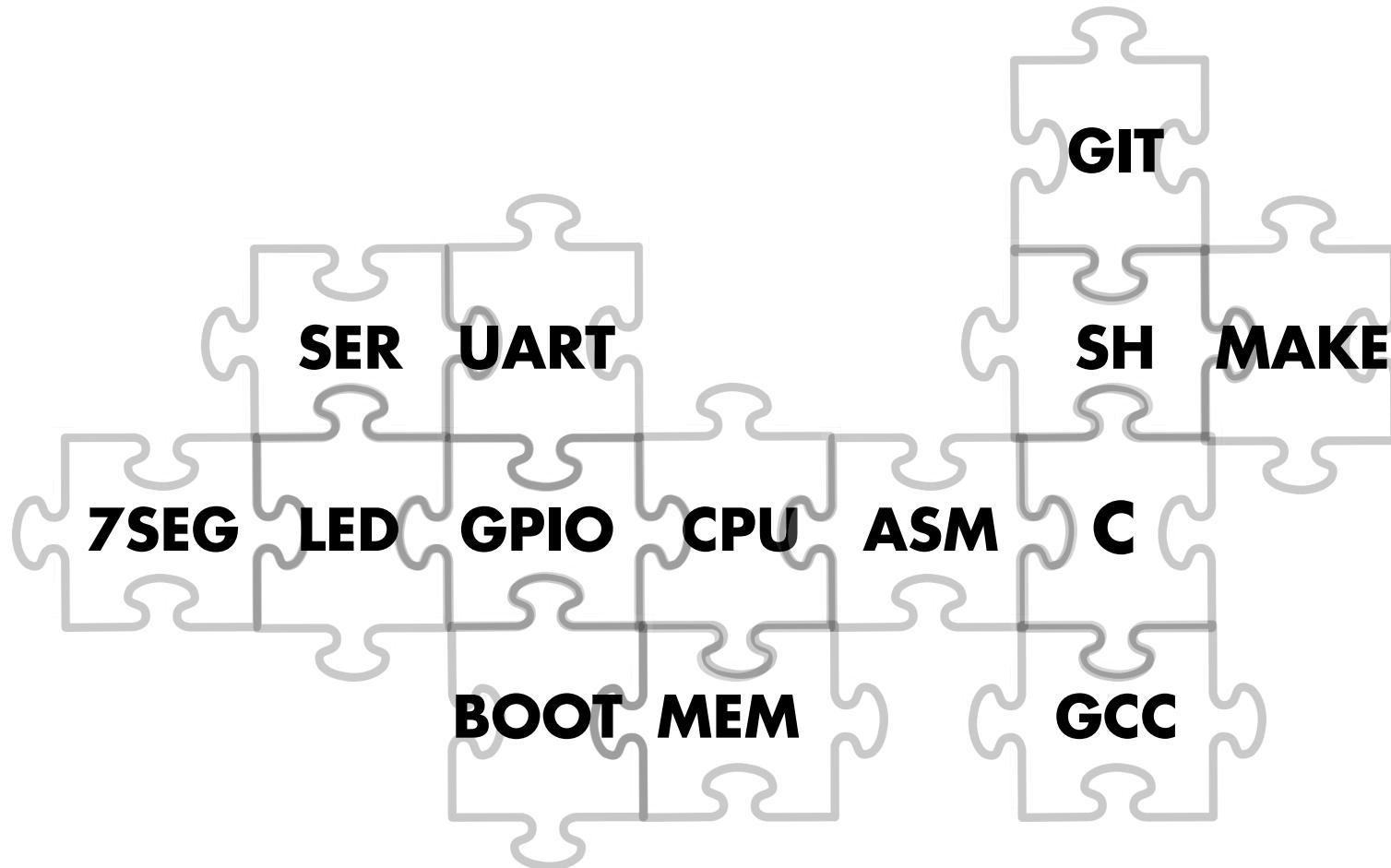
*

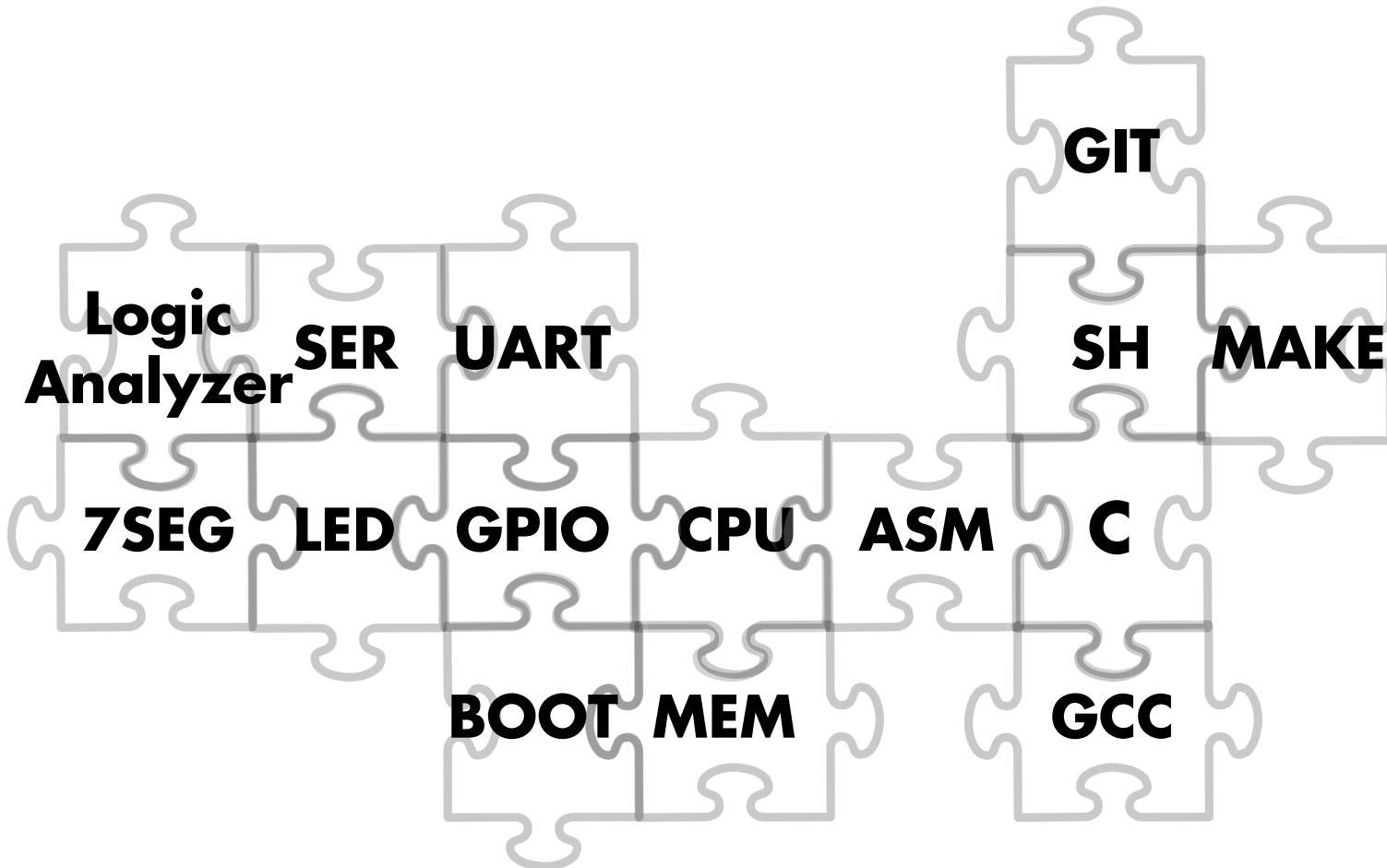
^

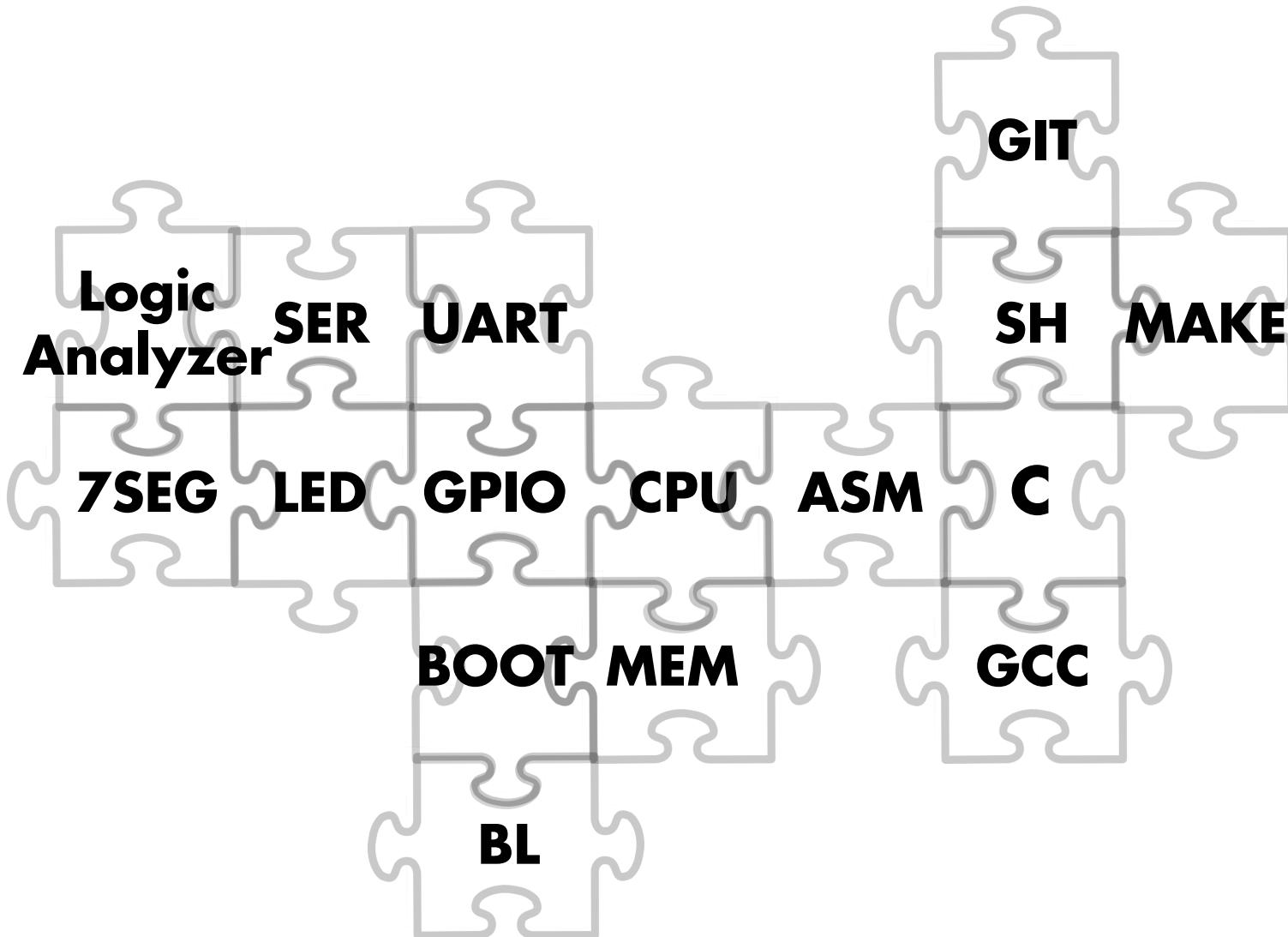
\$

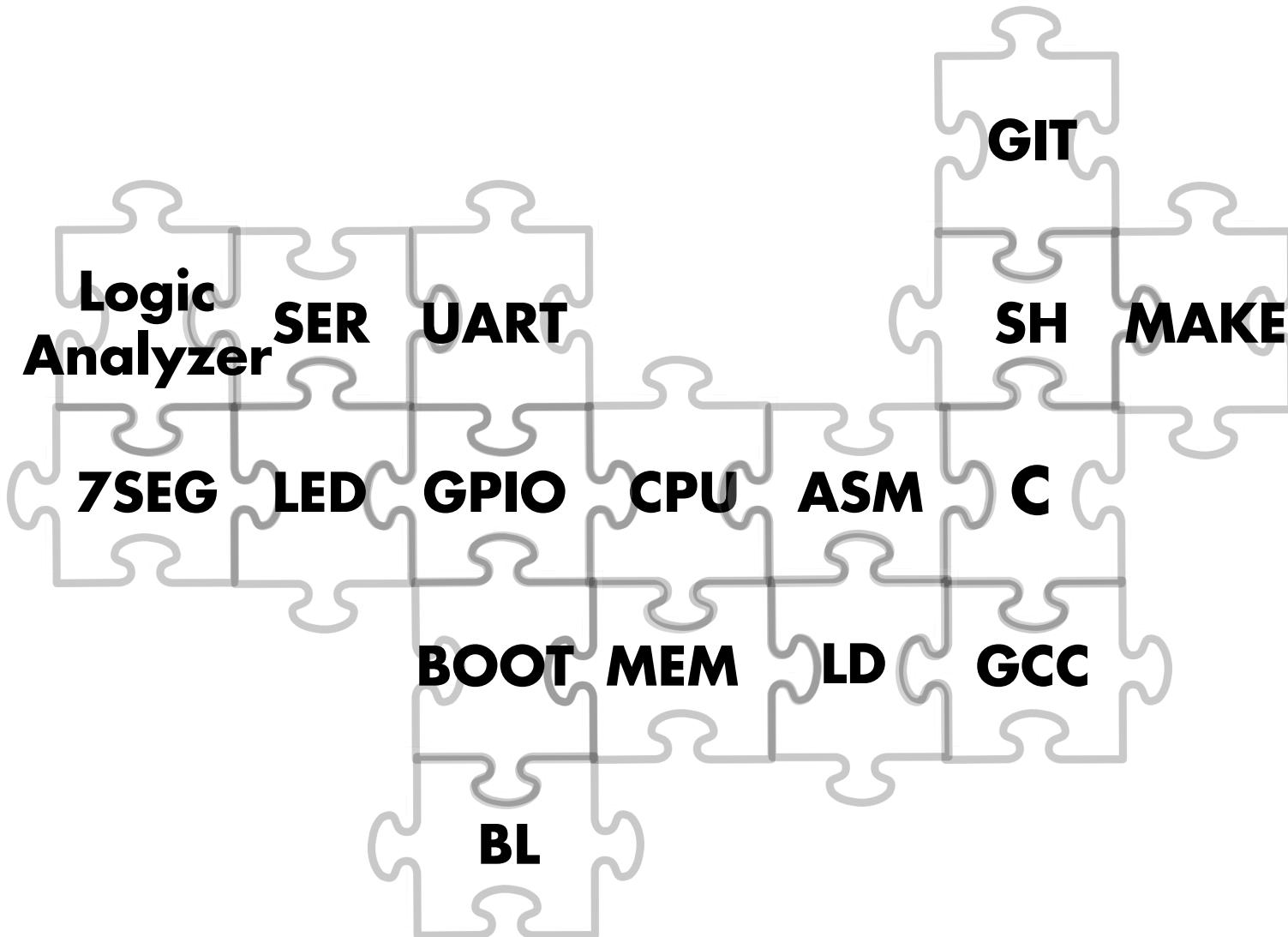
Read Beautiful Code

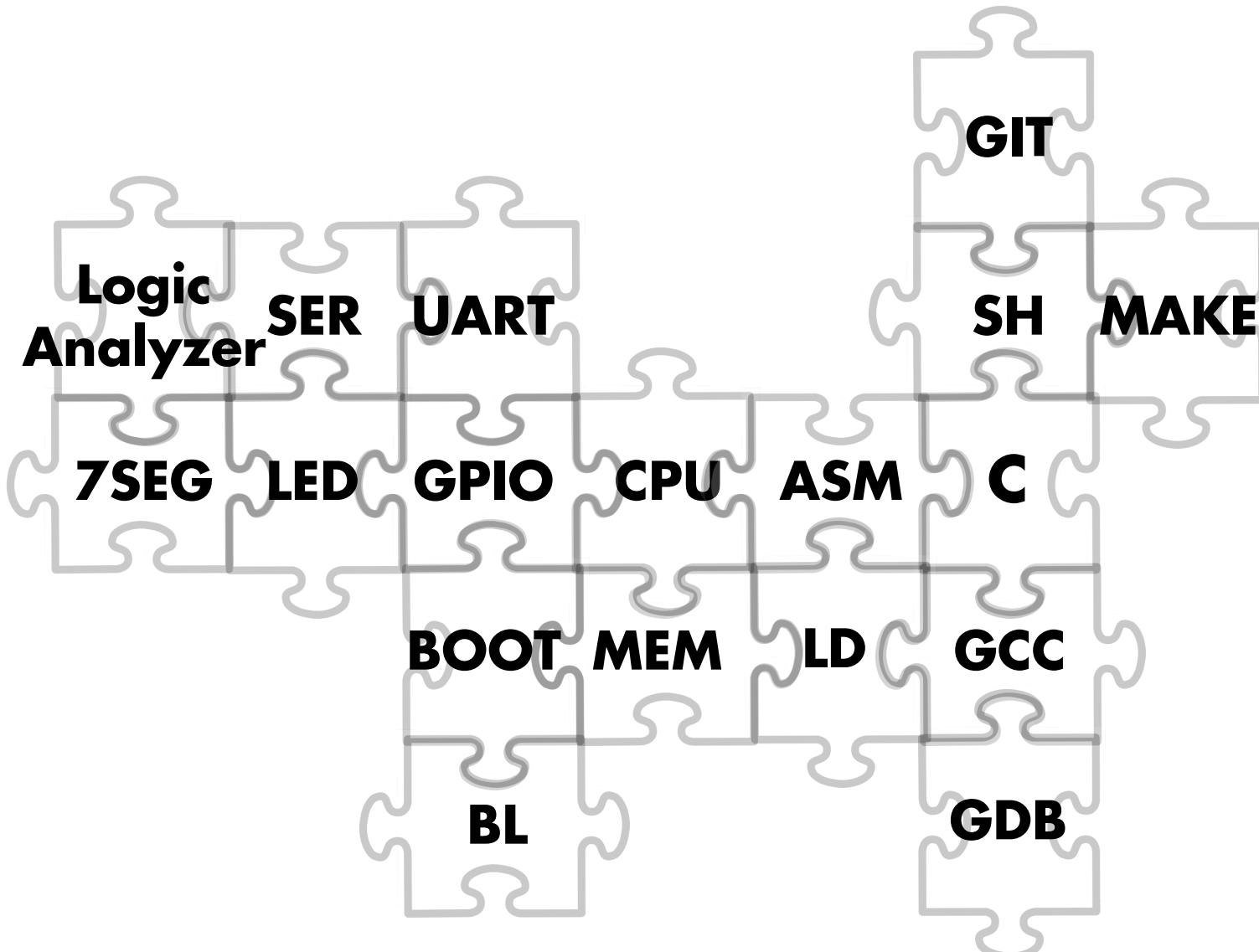


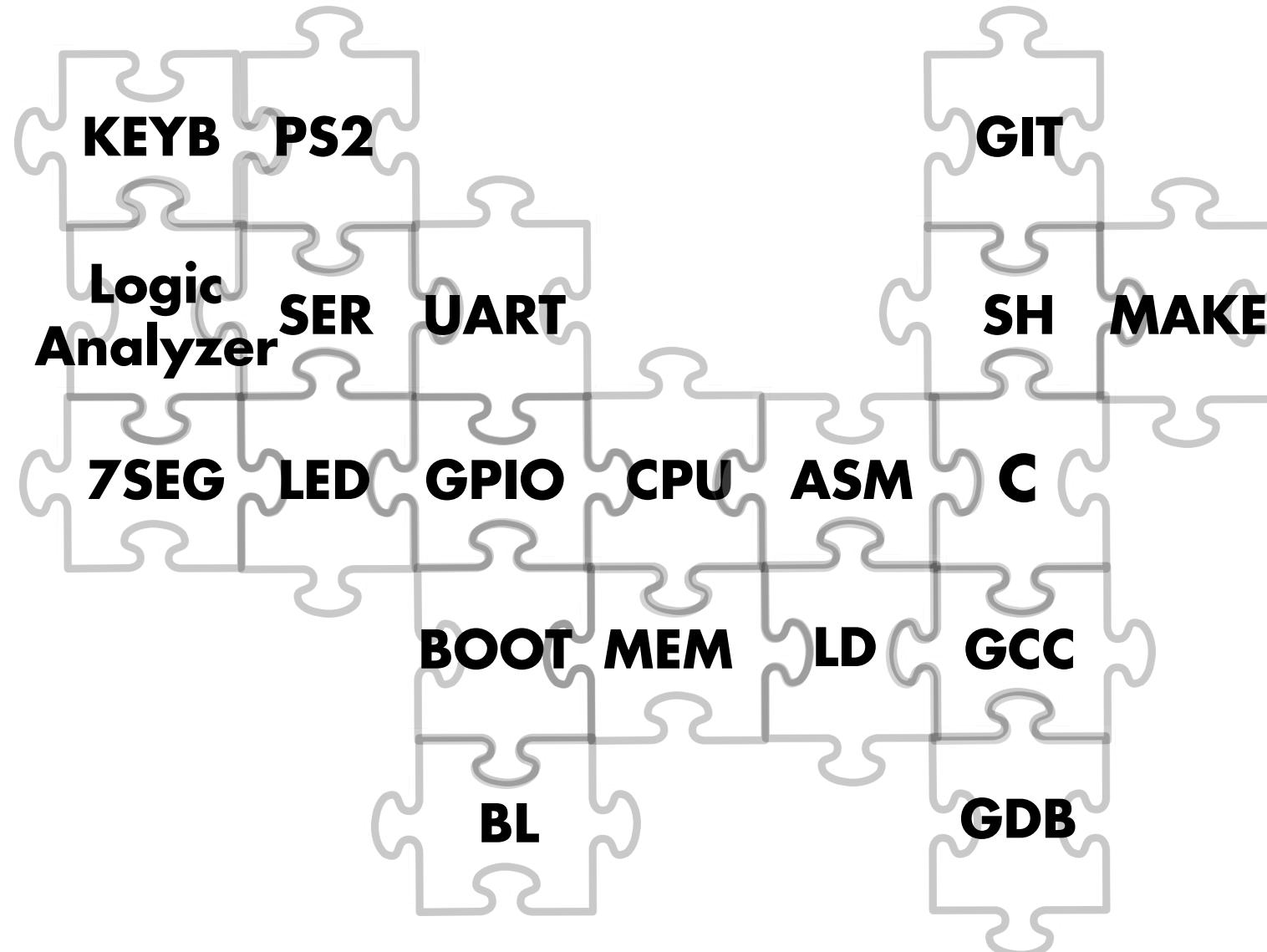


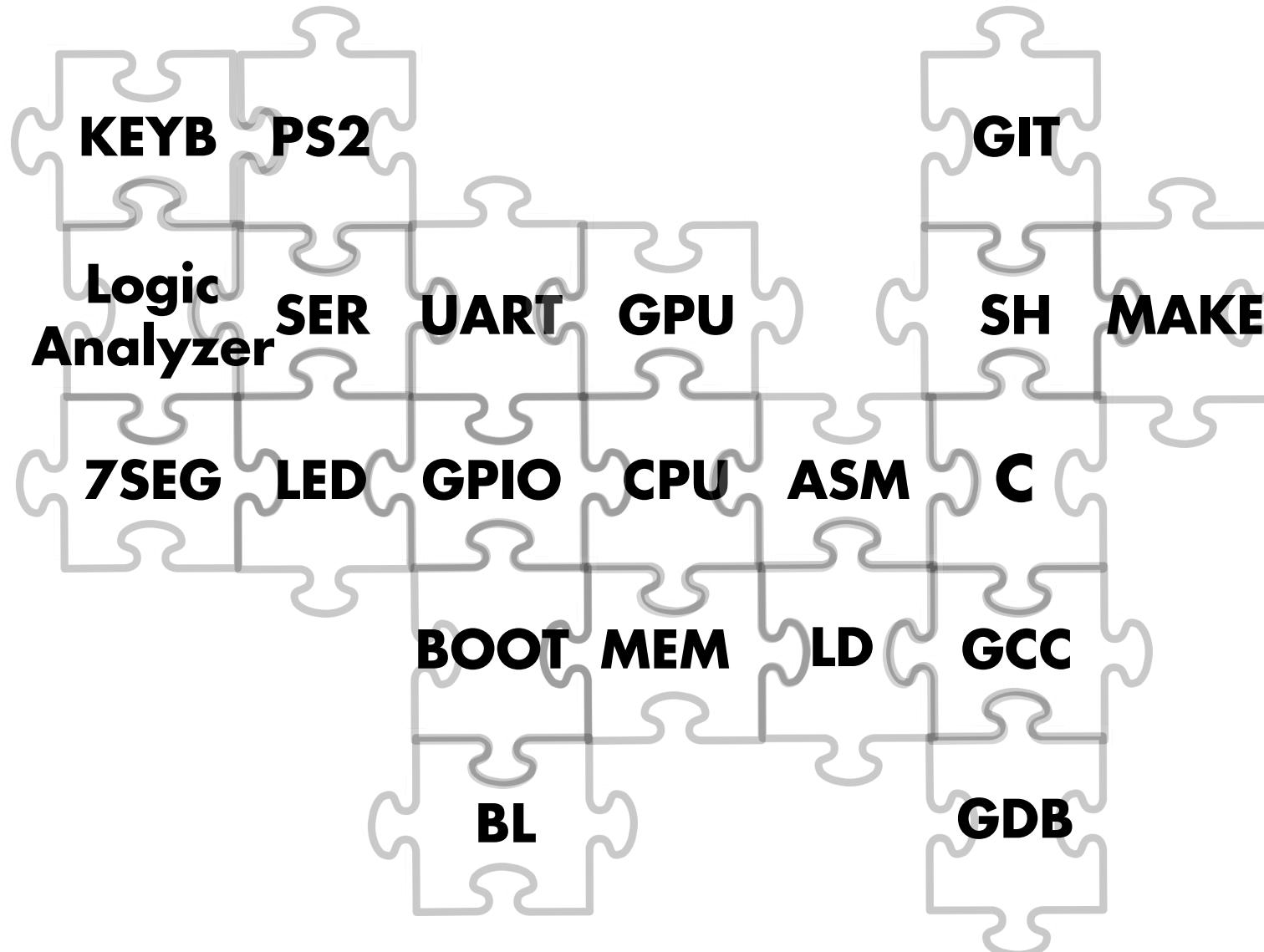




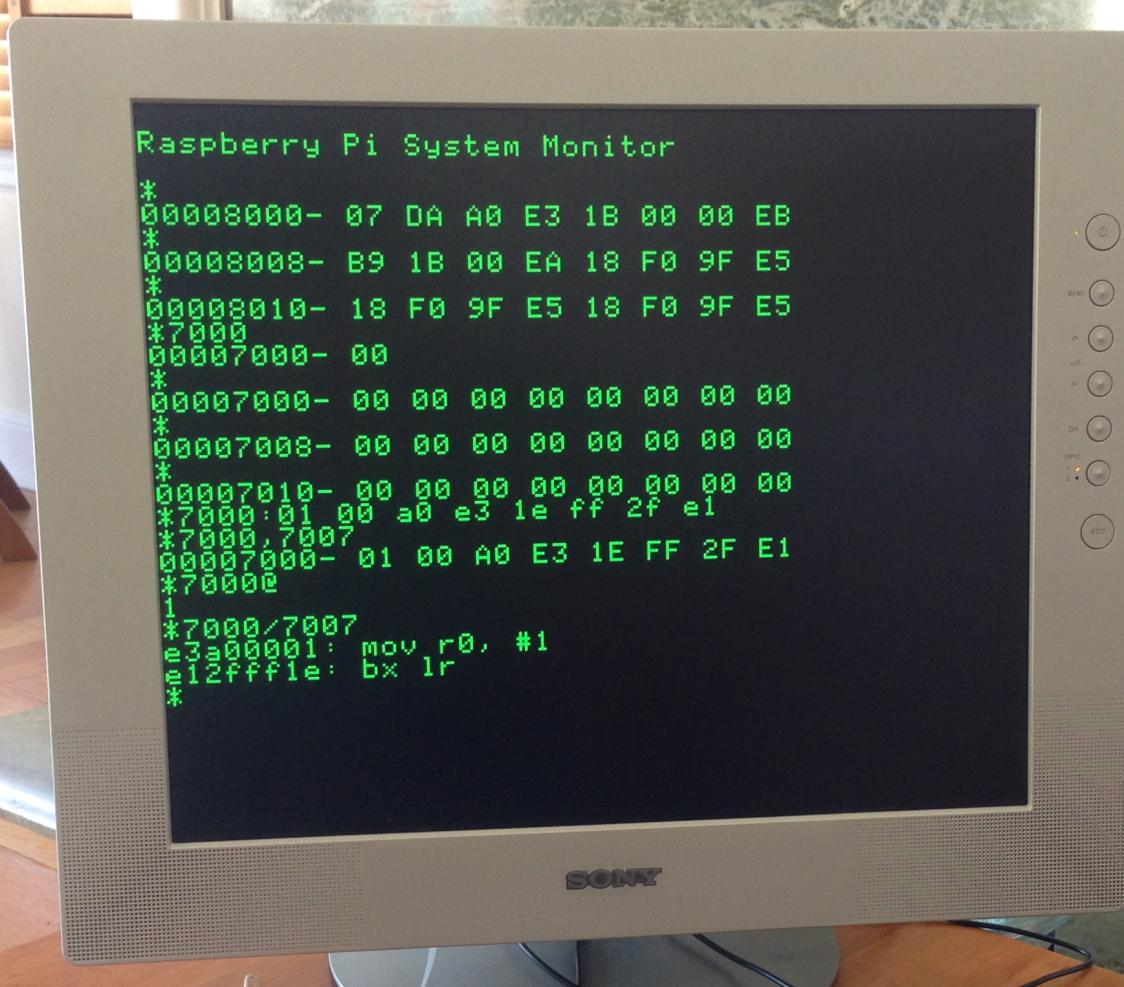
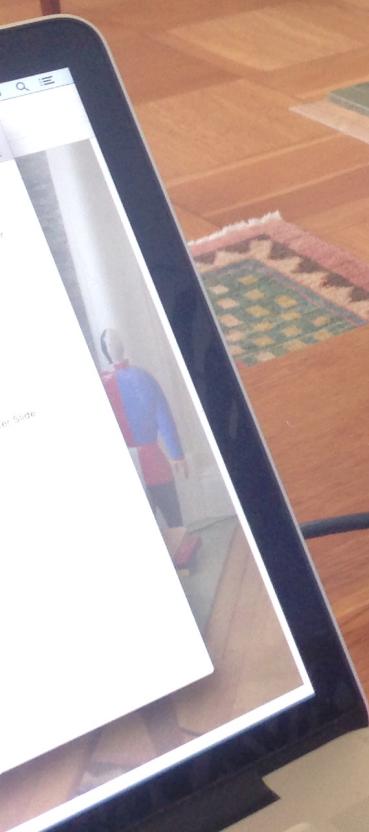






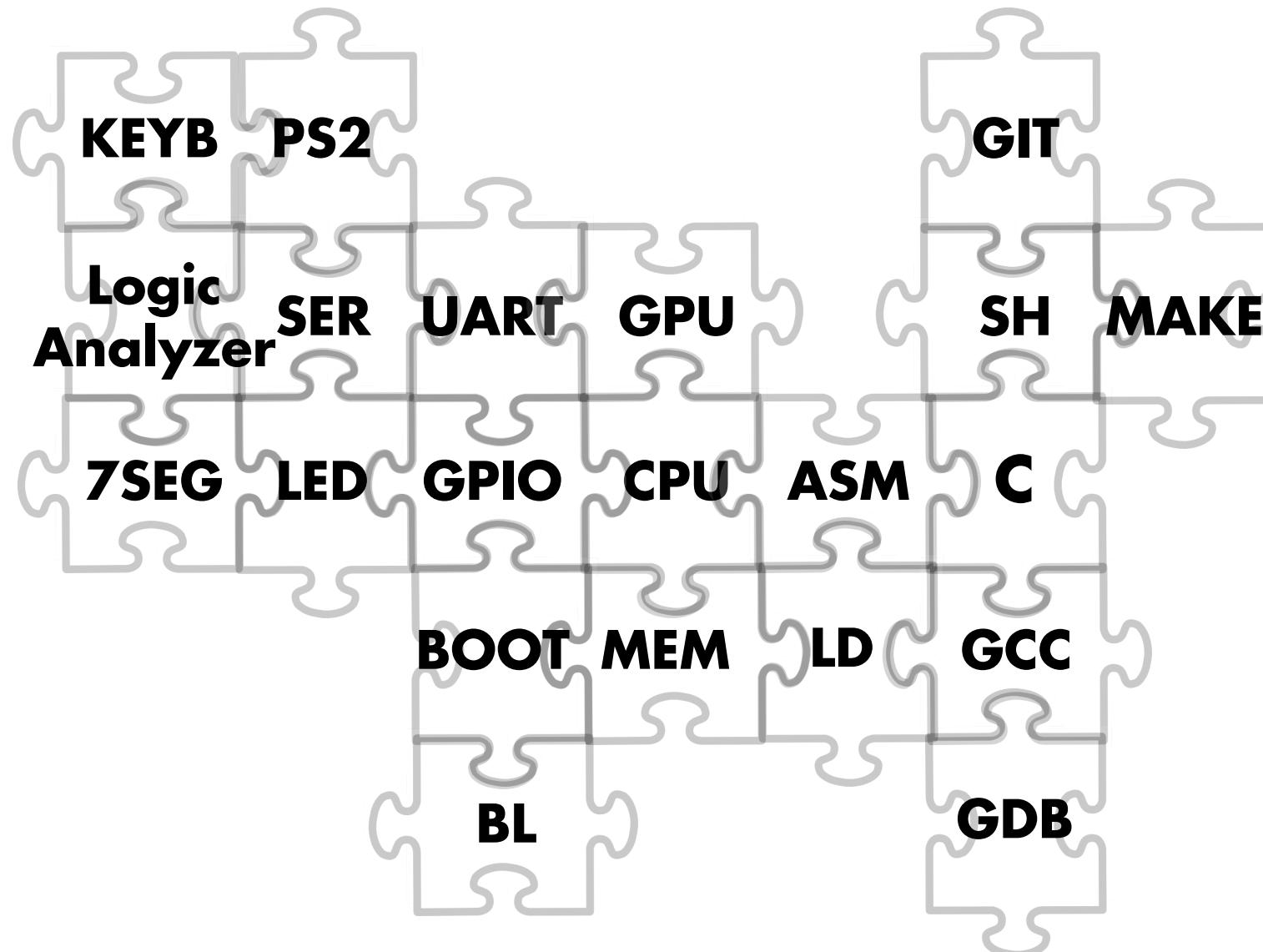


```
Raspberry Pi System Monitor  
*  
00008000- 07 DA A0 E3 1B 00 00 EB  
*  
00008008- B9 1B 00 EA 18 F0 9F E5  
*  
00008010- 18 F0 9F E5 18 F0 9F E5  
*7000  
00007000- 00  
*  
00007000- 00 00 00 00 00 00 00 00  
*  
00007008- 00 00 00 00 00 00 00 00  
*  
00007010- 00 00 00 00 00 00 00 00  
*7000:01 00 a0 e3 1e ff 2f e1  
*7000,7007  
00007000- 01 00 A0 E3 1E FF 2F E1  
*70000  
1  
*7000/7007  
e3a00001: mov r0, #1  
e12ffff1e: bx lr  
**
```

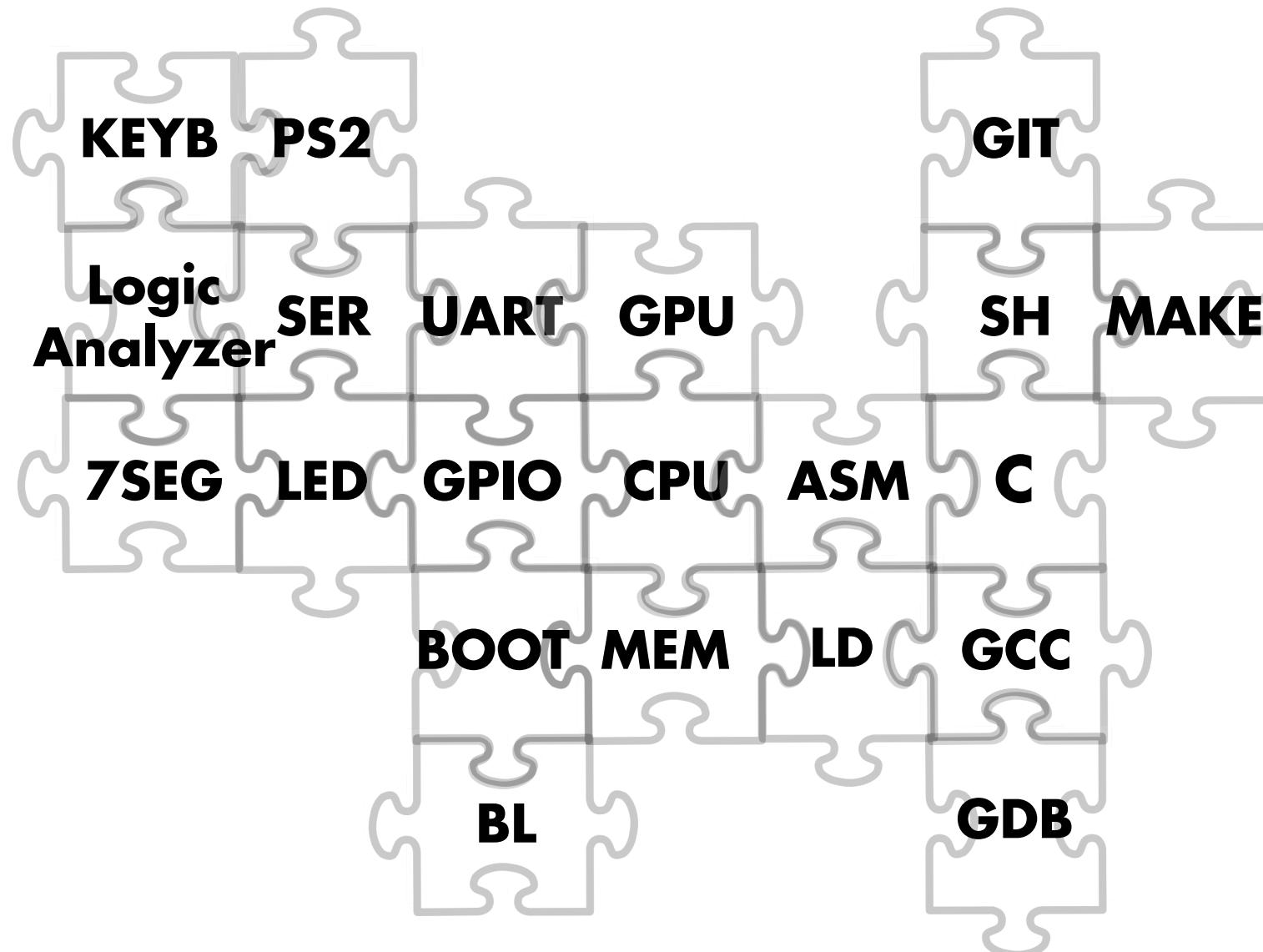


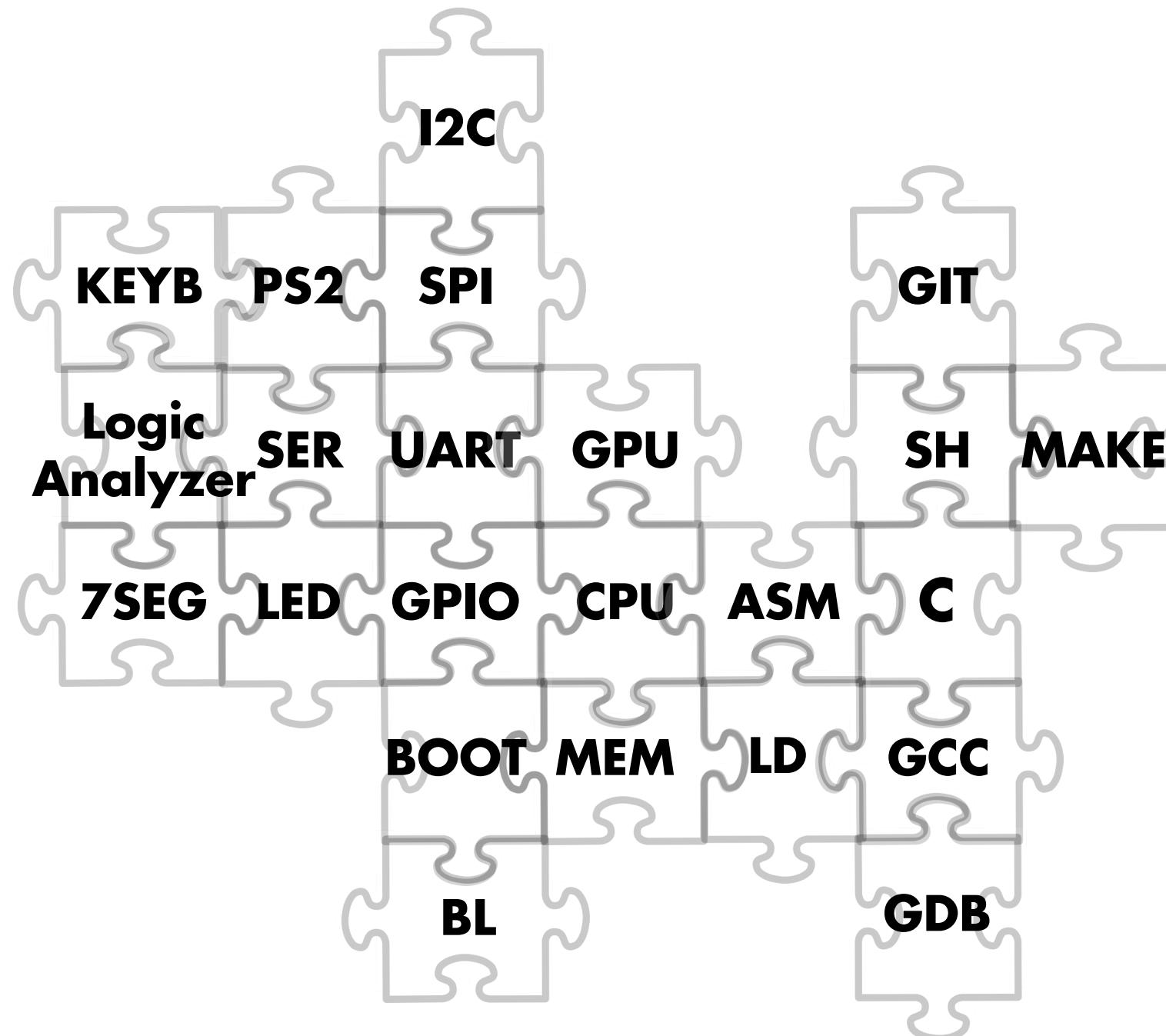
* *
* APPLE II *
* SYSTEM MONITOR *
* *
* COPYRIGHT 1977 BY *
* APPLE COMPUTER, INC. *
* *
* ALL RIGHTS RESERVED *
* *
* S. WOZNIAK *
* A. BAUM *
* *

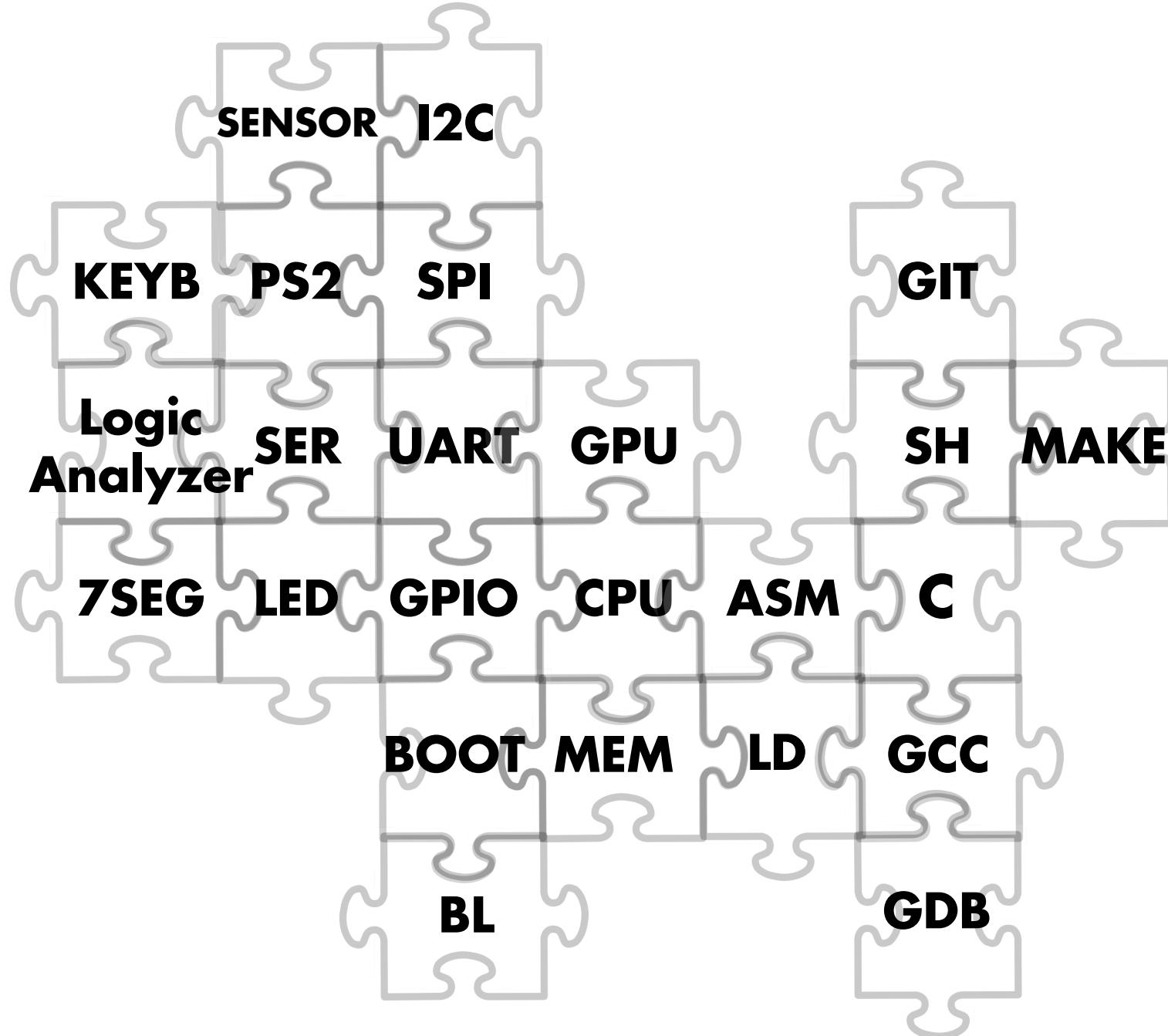
	TITLE	"APPLE II SYSTEM MONITOR"
LOC0	EPZ	\$00
LOC1	EPZ	\$01
WNDLFT	EPZ	\$20
WNDWDTH	EPZ	\$21
WNDTOP	EPZ	\$22
WNDBTM	EPZ	\$23
CH	EPZ	\$24
CV	EPZ	\$25
GBASL	EPZ	\$26
GBASH	EPZ	\$27
BASL	EPZ	\$28
BASH	EPZ	\$29
BAS2L	EPZ	\$2A
BAS2H	EPZ	\$2B
H2	EPZ	\$2C
LMNEM	EPZ	\$2C
RTNL	EPZ	\$2C
V2	EPZ	\$2D
RMNEM	EPZ	\$2D
RTNH	EPZ	\$2D
MASK	EPZ	\$2E

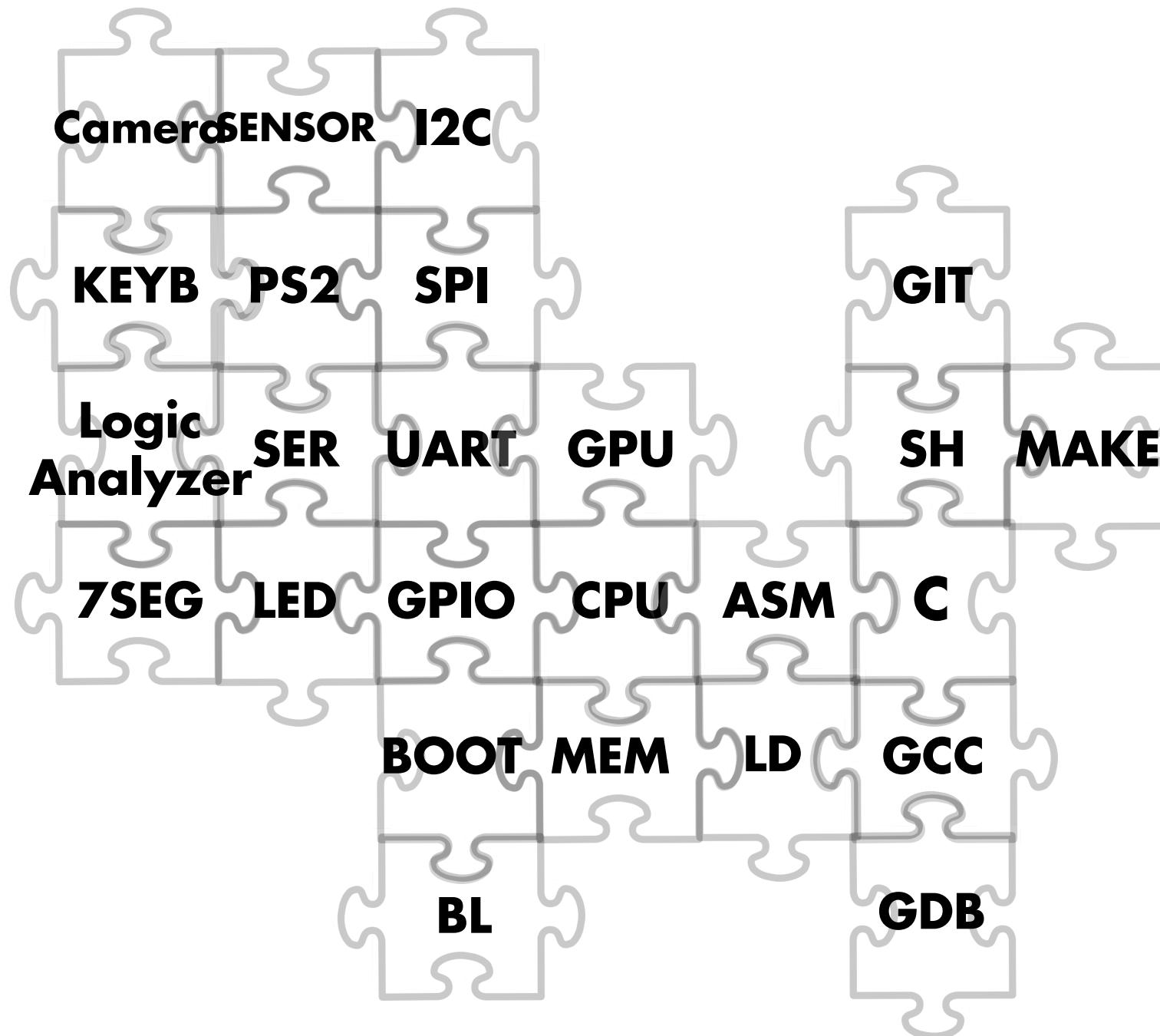


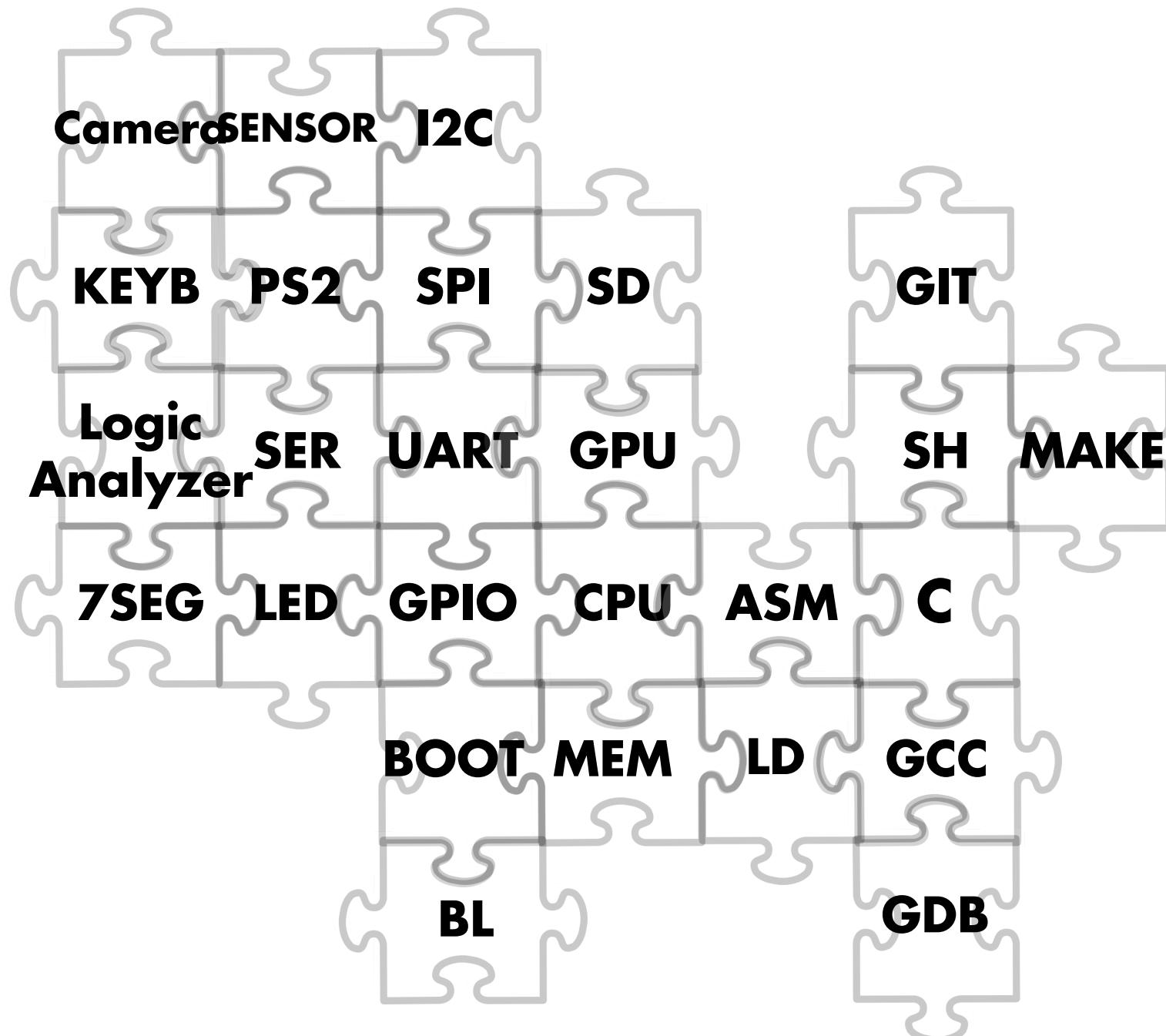
But It Keeps Growing!

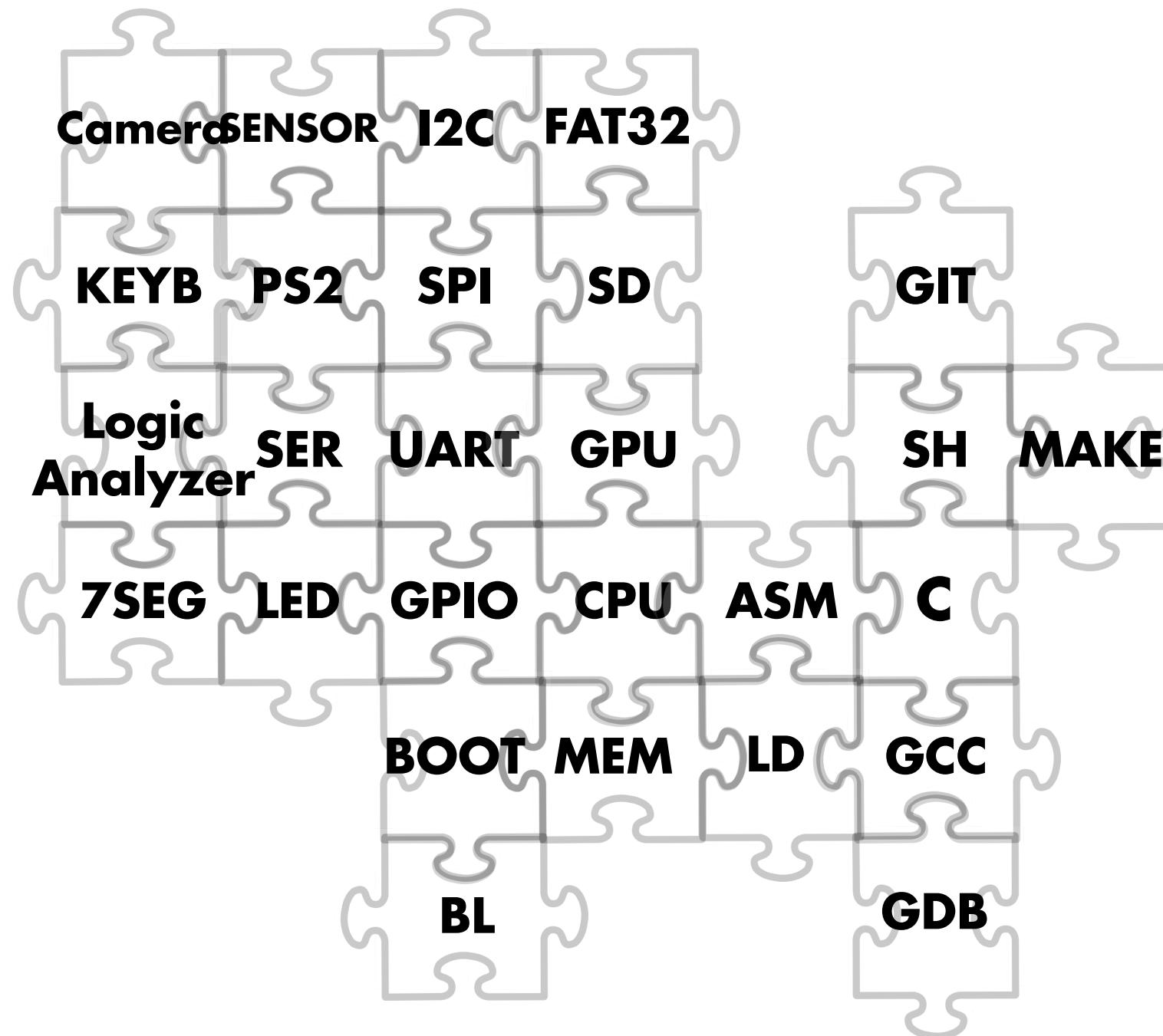


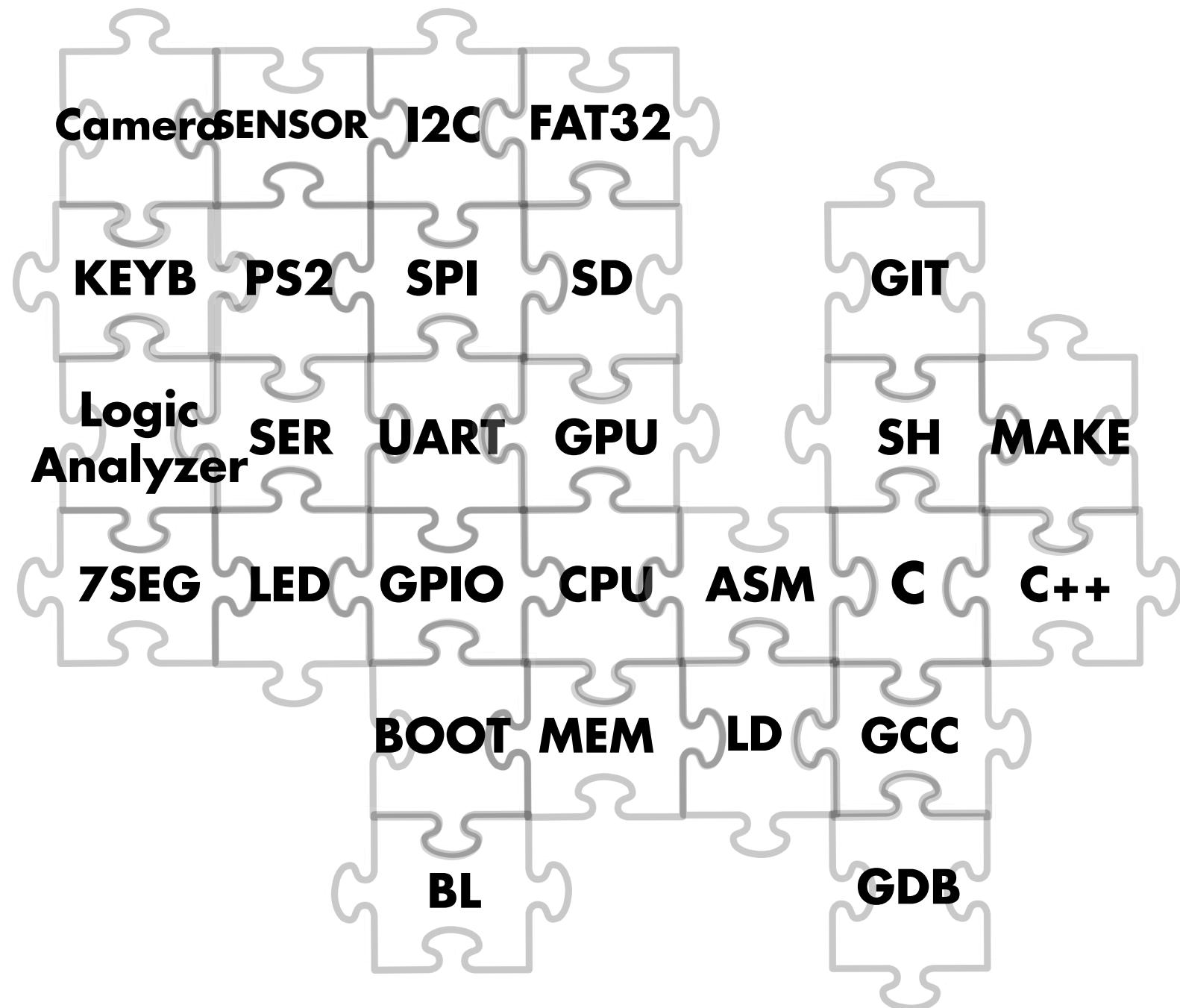


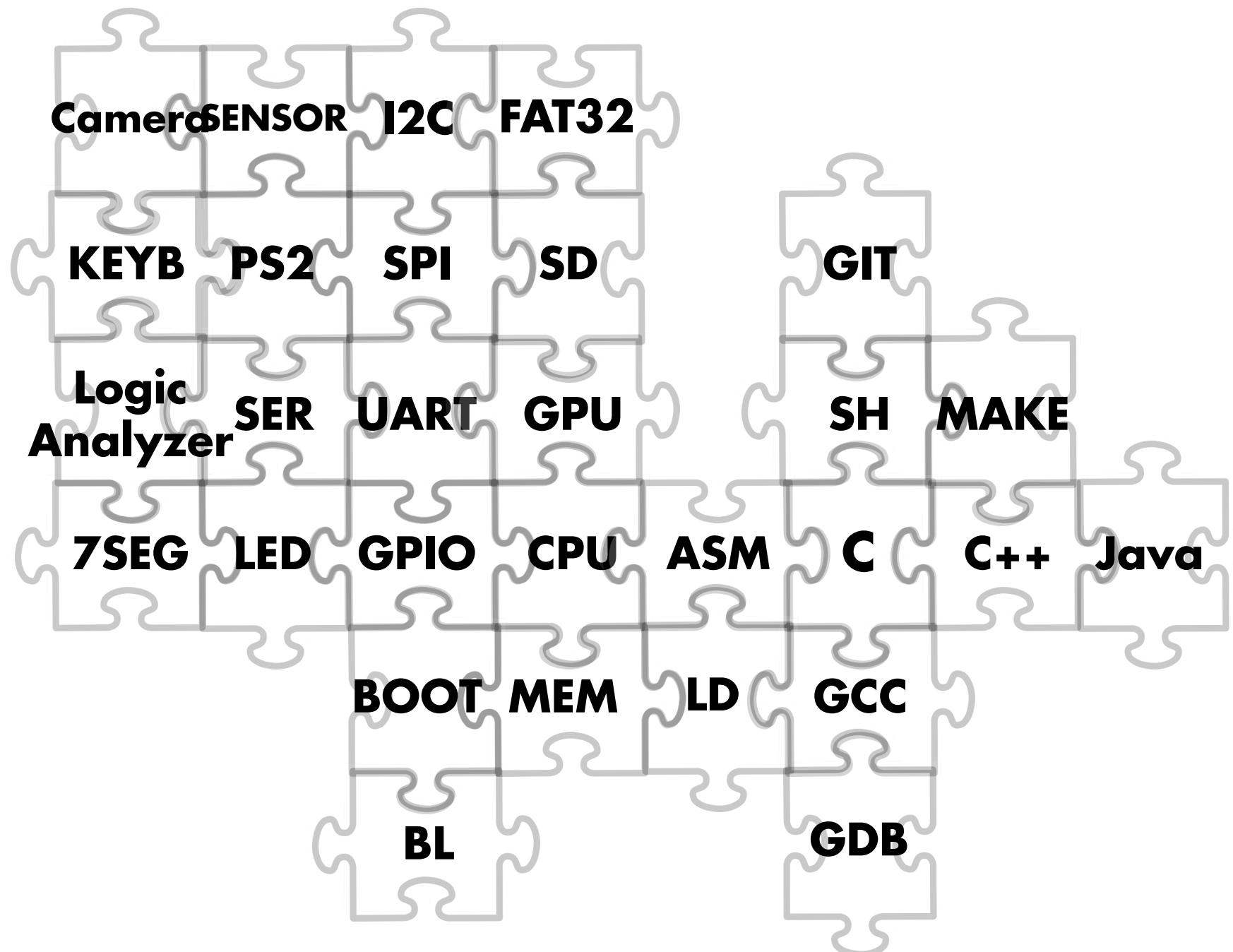


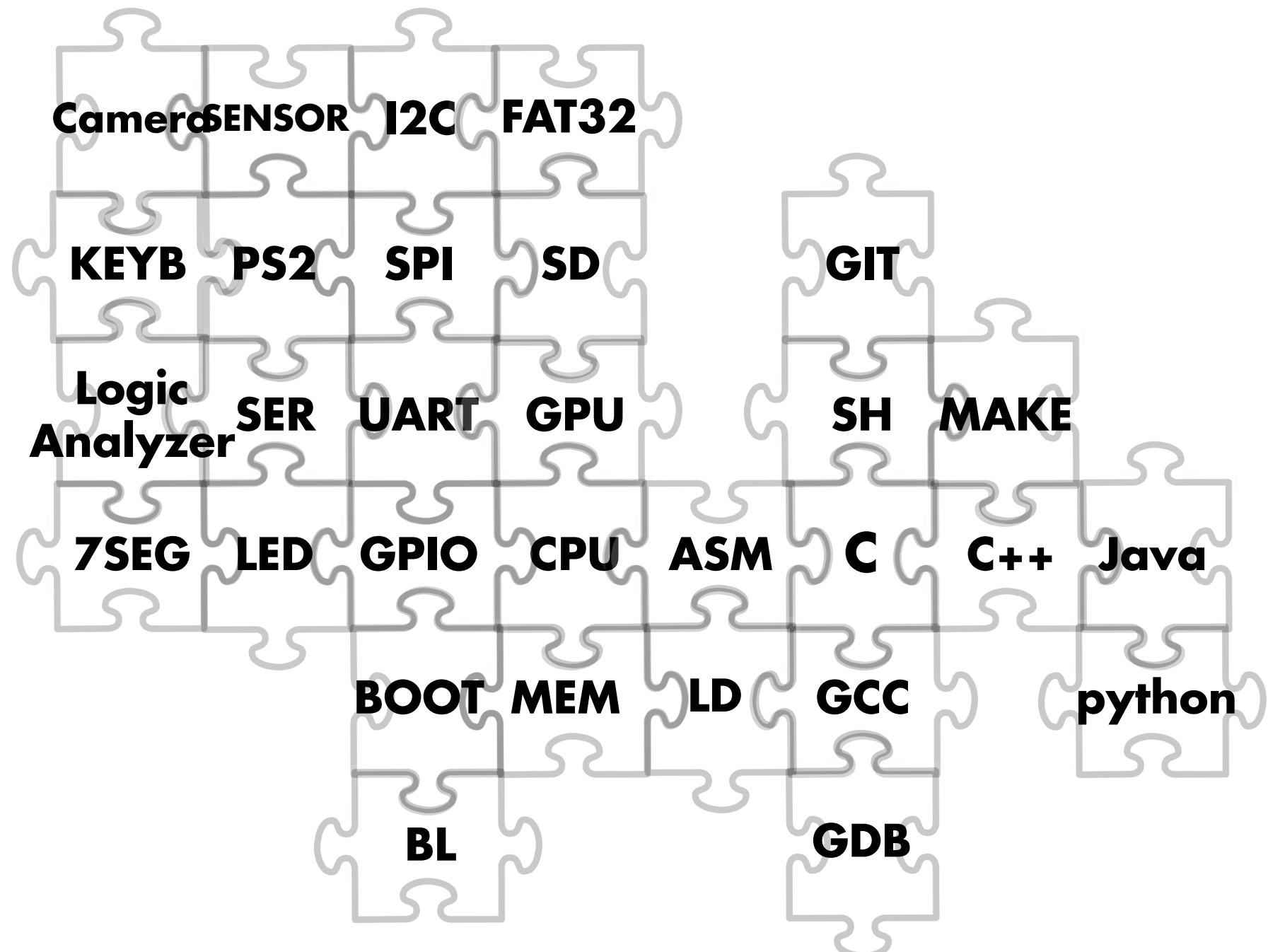






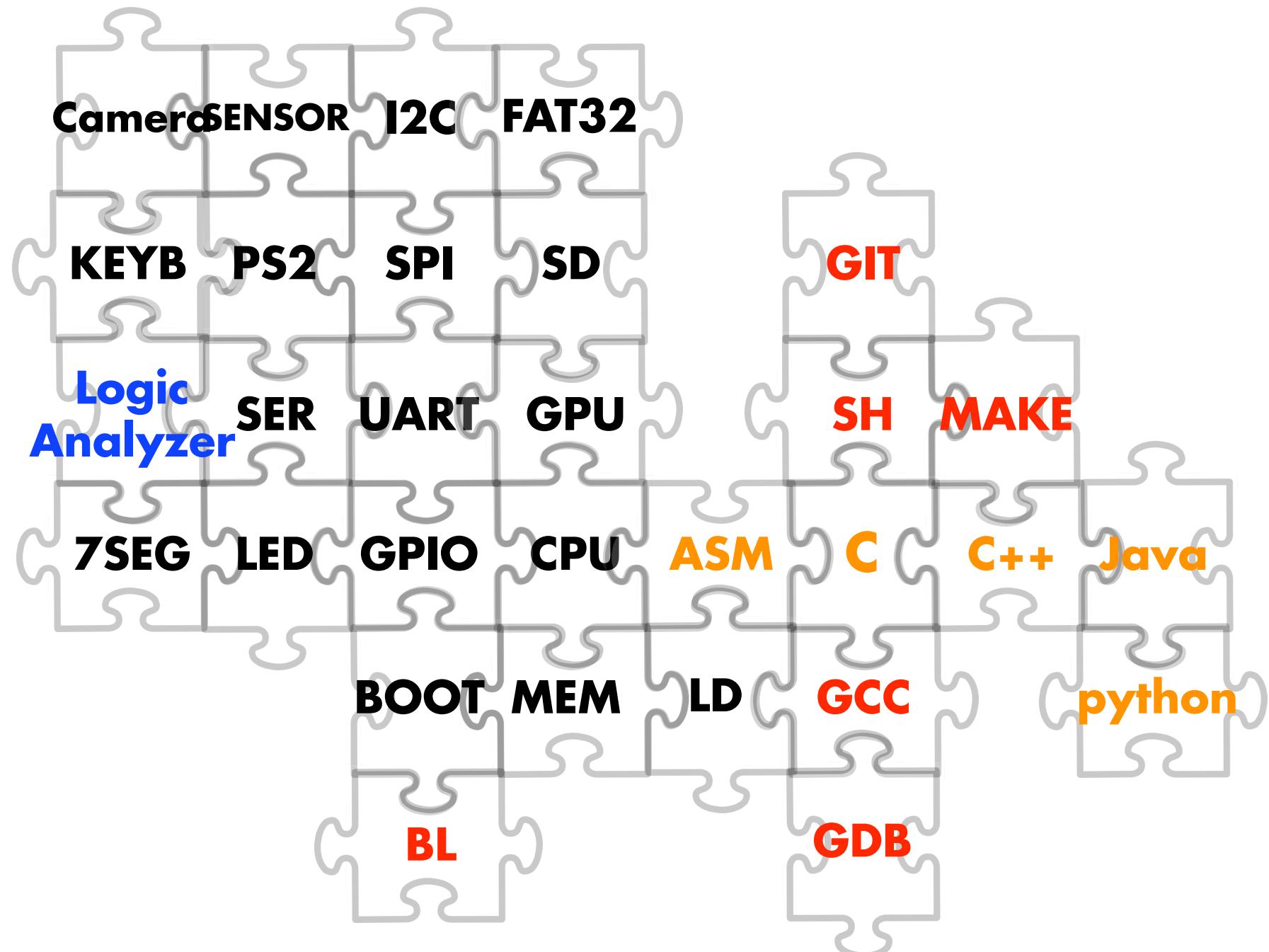


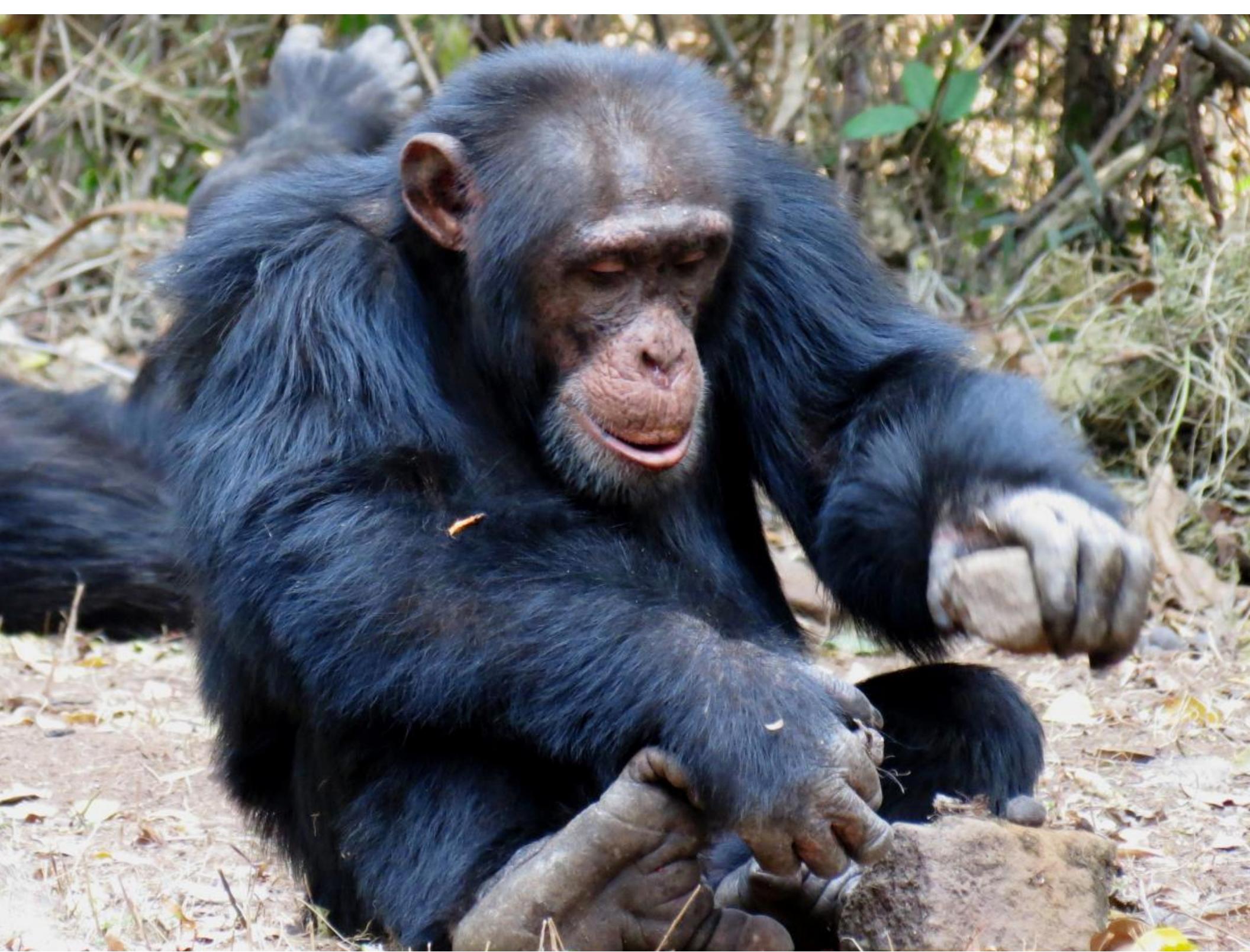




Goal 2

Master Your Tools









The Essential Set of Skills

Debugging, Testing, and Troubleshooting

Engineering Habits

Test, test, test, and test some more; Test as you go

Always start from a known working state, take small steps

Make things visible (printf, logic analyzer, gdb)

**Methodical (D&C), not random, search for solution.
Form hypotheses and perform experiments**

Fast prototyping, embrace automation, 1-click build

**Don't be frustrated by bugs, relish the challenge,
take frequent breaks**

Organized Development Environment



<http://amhistory.si.edu/juliachild/>



To invent, you need a good imagination and a pile of junk
Thomas Edison

Be a Maker and a Doer



Ren Ng invents the Light Field Camera



(A)



(B)

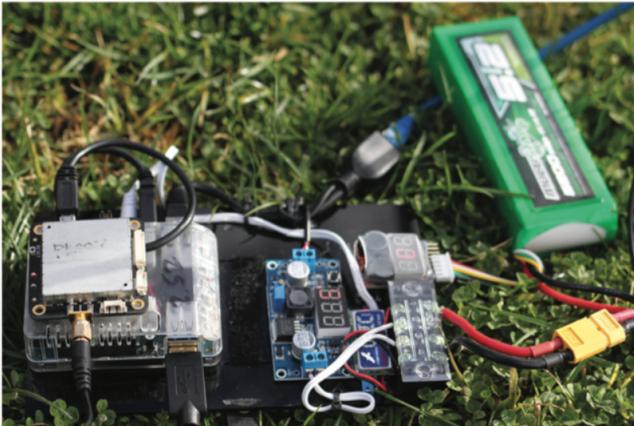
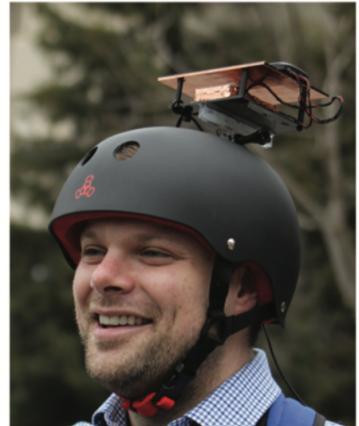


(C)

Figure 3.10: The medium format digital camera used in our prototype.

Lytro

Jane E, Niels Joubert, Mike Roberts



Linux and Beyond

Operating Systems

Multiple processes (scheduling)

Memory management (virtual memory)

Share peripherals (devices)

File systems and storage

Networking

Protect/secure processes from each other

Modes

		Privileged modes				
		Exception modes				
User	System	Supervisor	Abort	Undefined	Interrupt	Fast interrupt
R0	R0	R0	R0	R0	R0	R0
R1	R1	R1	R1	R1	R1	R1
R2	R2	R2	R2	R2	R2	R2
R3	R3	R3	R3	R3	R3	R3
R4	R4	R4	R4	R4	R4	R4
R5	R5	R5	R5	R5	R5	R5
R6	R6	R6	R6	R6	R6	R6
R7	R7	R7	R7	R7	R7	R7
R8	R8	R8	R8	R8	R8	R8_fiq
R9	R9	R9	R9	R9	R9	R9_fiq
R10	R10	R10	R10	R10	R10	R10_fiq
R11	R11	R11	R11	R11	R11	R11_fiq
R12	R12	R12	R12	R12	R12	R12_fiq
R13	R13	R13_svc	R13_abt	R13_und	R13_irq	R13_fiq
R14	R14	R14_svc	R14_abt	R14_und	R14_irq	R14_fiq
PC	PC	PC	PC	PC	PC	PC

CPSR	CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
		SPSR_svc	SPSR_abt	SPSR_und	SPSR_irq	SPSR_fiq

△ indicates that the normal register used by User or System mode has been replaced by an alternative register specific to the exception mode

Modes

User	System	Supervisor	Abort	Undefined	Interrupt	Fast interrupt
R0	R0	R0	R0	R0	R0	R0
R1	R1	R1			R1	R1
R2	R2	R2	H2	H2	R2	R2
R3	R3	R3	R3	R3	R3	R3
R4	R4	R4	R4	R4	R4	R4
R5	R5	R5	R5	R5	R5	R5
R6	R6	R6	R6	R6	R6	R6
R7	R7	R7	R7	R7	R7	R7
R8	R8	R8	R8	R8	R8	R8_fiq
R9	R9	R9	R9	R9	R9	R9_fiq
R10	R10	R10	R10	R10	R10	R10_fiq
R11	R11	R11	R11	R11	R11	R11_fiq
R12	R12	R12	R12	R12	R12	R12_fiq
R13	R13	R13_svc	R13_abt	R13_und	R13_irq	R13_fiq
R14	R14	R14_svc	R14_abt	R14_und	R14_irq	R14_fiq
PC	PC	PC	PC	PC	PC	PC
CPSR	CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
		SPSR_svc	SPSR_abt	SPSR_und	SPSR_irq	SPSR_fiq

△ indicates that the normal register used by User or System mode has been replaced by an alternative register specific to the exception mode

Modes

		Privileged modes				
		Exception modes				
User	System	Supervisor	Abort	Undefined	Interrupt	Fast interrupt
R0	R0	R0	R0	R0	R0	R0
R1	R1	R1			R1	R1
R2	R2	R2	H2	H2	R2	R2
R3	R3	R3	R3	R3	R3	R3
R4	R4	R4	R4	R4	R4	R4
R5	R5	R5	R5	R5	R5	R5
R6	R6	R6	R6	R6	R6	R6
R7	R7	R7	R7	R7	R7	R7
R8	R8	R8	R8	R8	R8	R8_fiq
R9	R9	R9	R9	R9	R9	
R10	R10	R10	R10	R10	R10	
R11	R11	R11	R11	R11	R11	R11_fiq
R12	R12	R12	R12	R12	R12	R12_fiq
R13	R13	R13_svc	R13_abt	R13_und	R13_irq	R13_fiq
R14	R14	R14_svc	R14_abt	R14_und	R14_irq	R14_fiq
PC	PC	PC	PC	PC	PC	PC
CPSR	CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
		SPSR_svc	SPSR_abt	SPSR_und	SPSR_irq	SPSR_fiq



indicates that the normal register used by User or System mode has been replaced by an alternative register specific to the exception mode

Interrupt code

Modes

		Privileged modes				
		Exception modes				
User	System	Supervisor	Abort	Undefined	Interrupt	Fast interrupt
R0	R0	R0	R0	R0	R0	R0
R1	R1	R1			R1	R1
R2	R2	R2	H2	H2	R2	R2
R3	R3	R3	R3	R3	R3	R3
R4	R4	R4	R4	R4	R4	R4
R5	R5	R5	R5	R5	R5	R5
R6	R6	R6	R6	R6	R6	R6
R7	R7	R7	R7	R7	R7	R7
R8	R8	R8	R8	R8	R8	R8_fiq
R9	R9	R9	R9	R9	R9	
R10	R10	R10	R10	R10	R10	
R11	R11	R11	R11	R11	R11	R11_fiq
R12	R12	R12	R12	R12	R12	R12_fiq
R13	R13	R13_svc	R13_abt	R13_und	R13_irq	R13_fiq
R14	R14	R14_svc	R14_abt	R14_und	R14_irq	R14_fiq
PC	PC	PC	PC	PC	PC	PC
CPSR	CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
		SPSR_svc	SPSR_abt	SPSR_und	SPSR_irq	SPSR_fiq



indicates that the normal register used by User or System mode has been replaced by an alternative register specific to the exception mode

Interrupt code

User code

Bare metal code

User vs Supervisor Mode

User mode can't do a lot of things ...

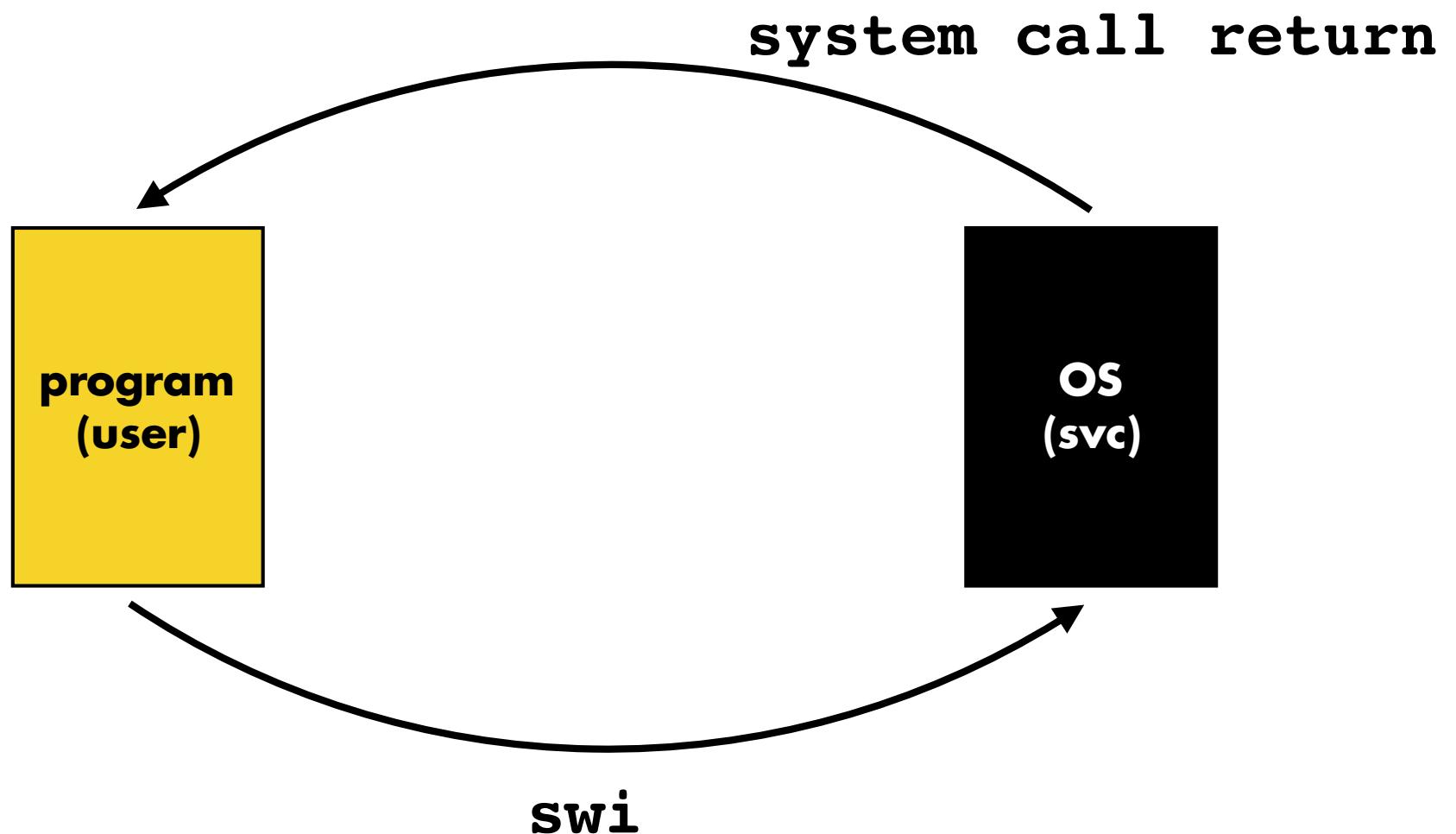
- 1. Modify CPSR (change modes)**
- 2. Access configuration registers**
- 3. Access IO registers/memory**

Hardware throws an exception if user mode program does any of these things

Enables operating system to keep processes from interfering with each other

Essential for security

Operating System



Next Steps

EE108

Digital Systems

EE180

Digital Architecture

CS240

Operating Systems

CS110

Computer Systems

CS142

Web Applications

CS140

Operating Systems

CS143

Compilers

CS144

Networks

CS145

Databases

CS148

Graphics

CS149

Parallel Computing

CS241

Embed. Wrkshp.

CS242

Prog. Langs.

CS243

Compilers

CS244

Networks

CS245

Databases

CS248

Graphics

Labs this week

- Milestones!**
- Make sure you have all your parts**

Demonstrations

- **Tue June 7th from 3:30 to 6:30 pm in Gates 325**
- **Start with fast forward : 2 minute project pitch**
- **Setup project and demo to students and instructors**

Submission

- **Due Wed June 8th at midnight**
- **Final github commit of code**
- **Include README.md describing your project (pictures!!, attributions)**

Logistics

- **Return keyboards (the rest of the stuff is yours)**
- **Submit reimbursements**

Comments and Suggestions?