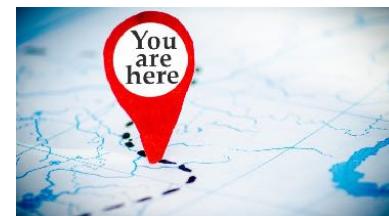


CS107e Wrap Party

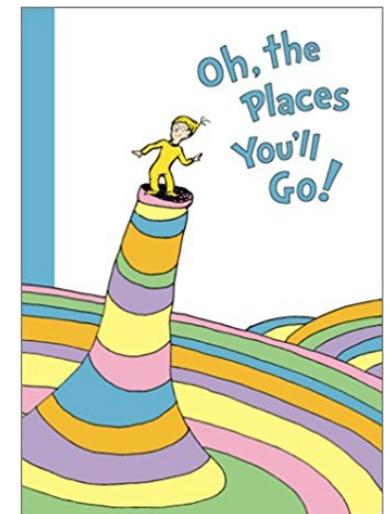


Where we started



How far we've come

Where to go next



Project demos

Fri Dec 13 9-11am in Gates B02

Each team has 3-4 minutes to give pitch and show demo

Remaining time to circulate and check out projects

Project submission

Due end of day **Fri Dec 13**, push all to GitHub

Include README.md (description, pictures, attributions)

End-quarter logistics

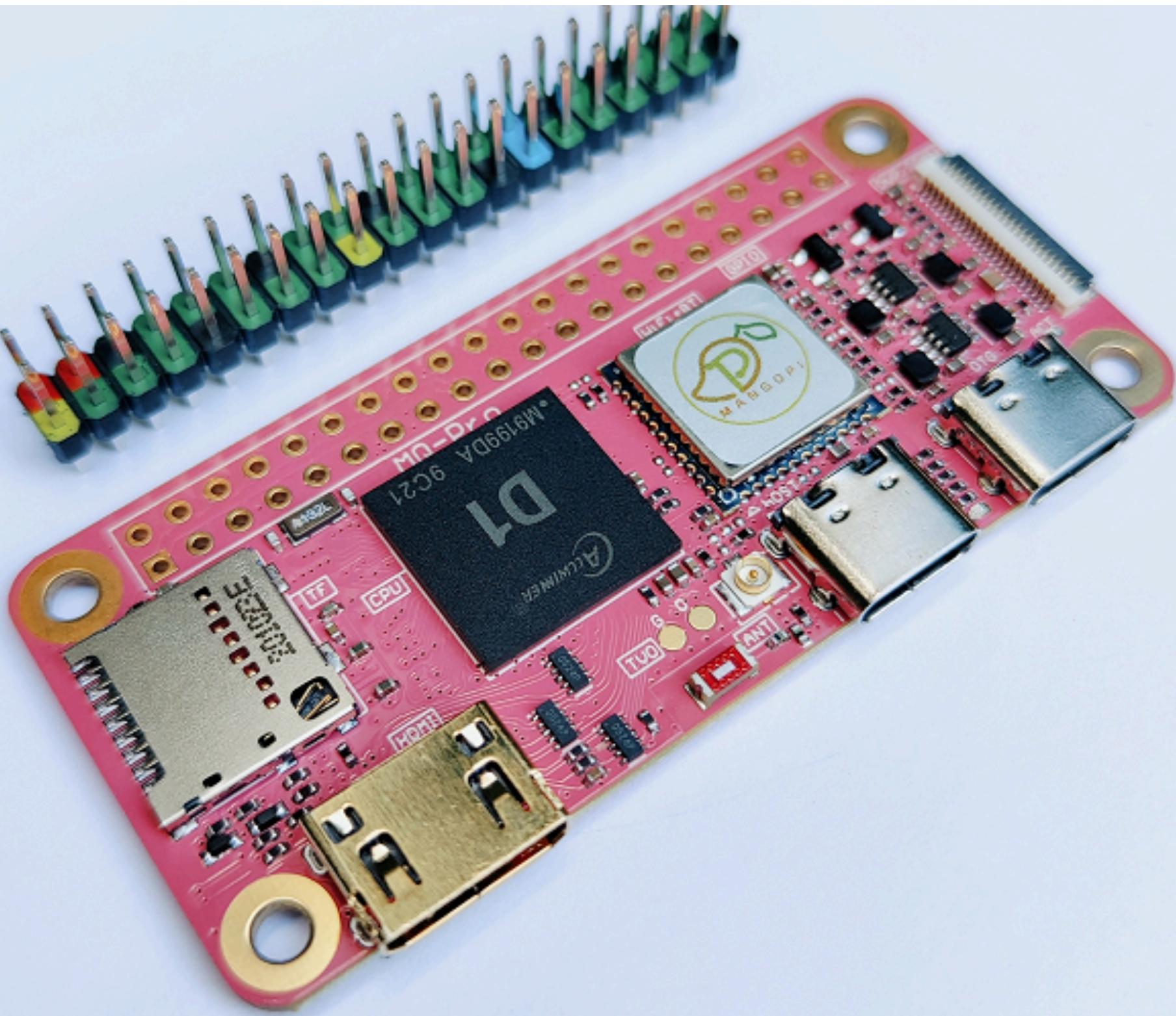
Return keyboard + PS2 plugboard

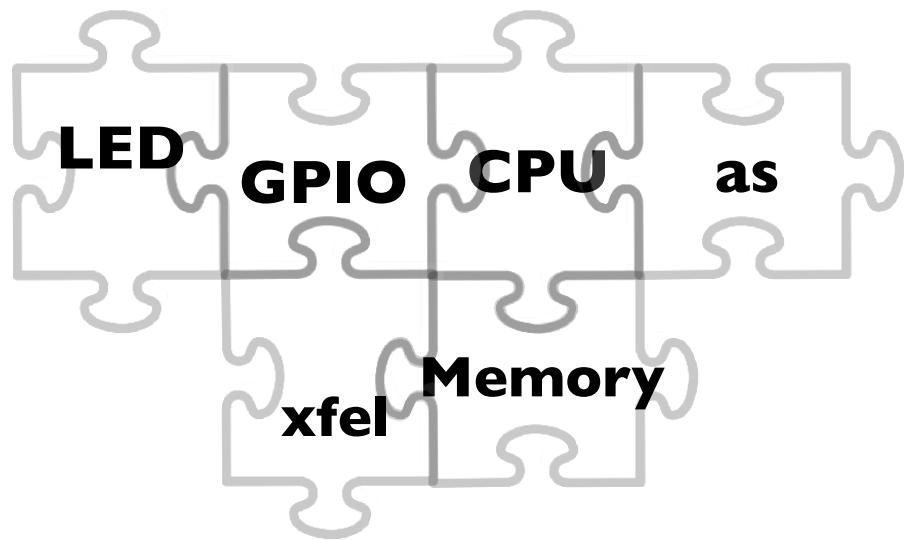
If not using for final project, return this week; otherwise at demo day

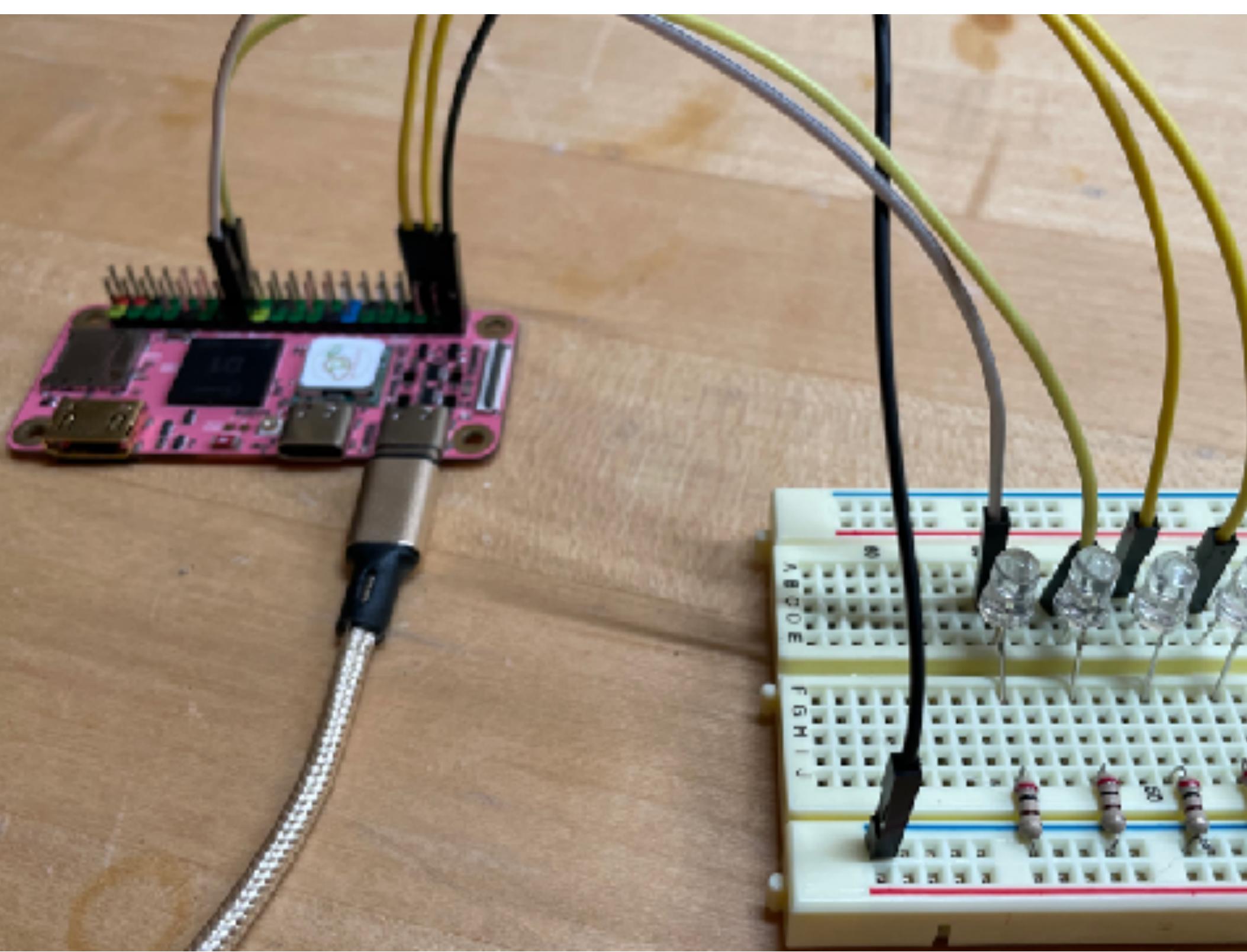
Please fill out course evaluation, offer feedback for future offerings

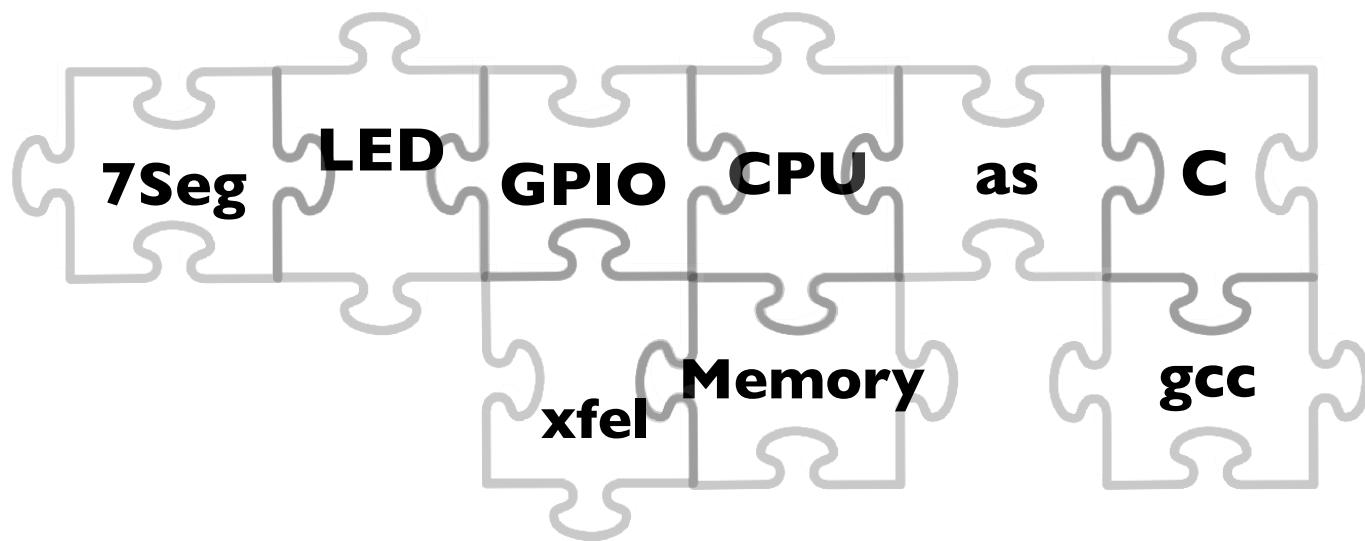
Course learning goal #1:

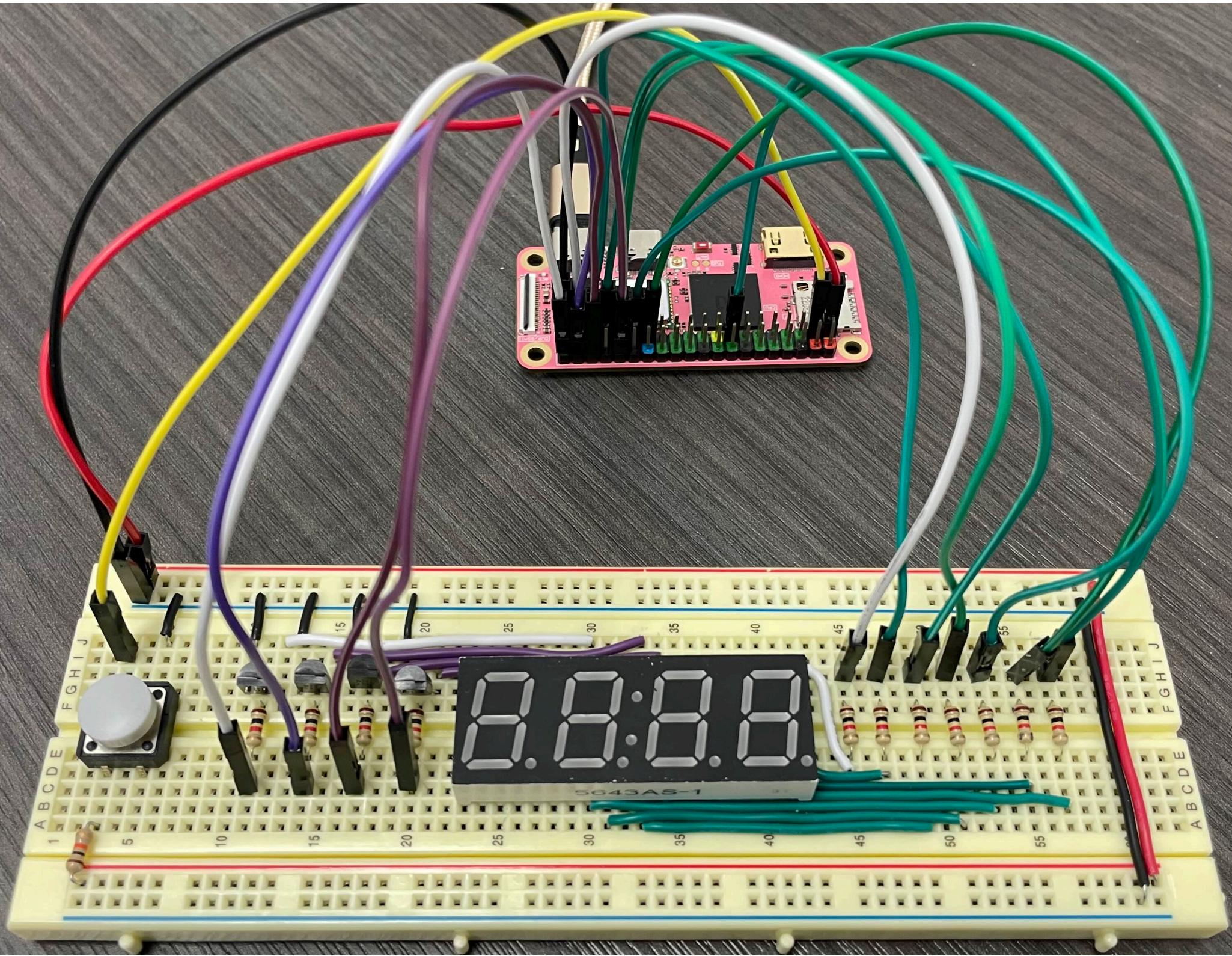
**Understand how
computers work**

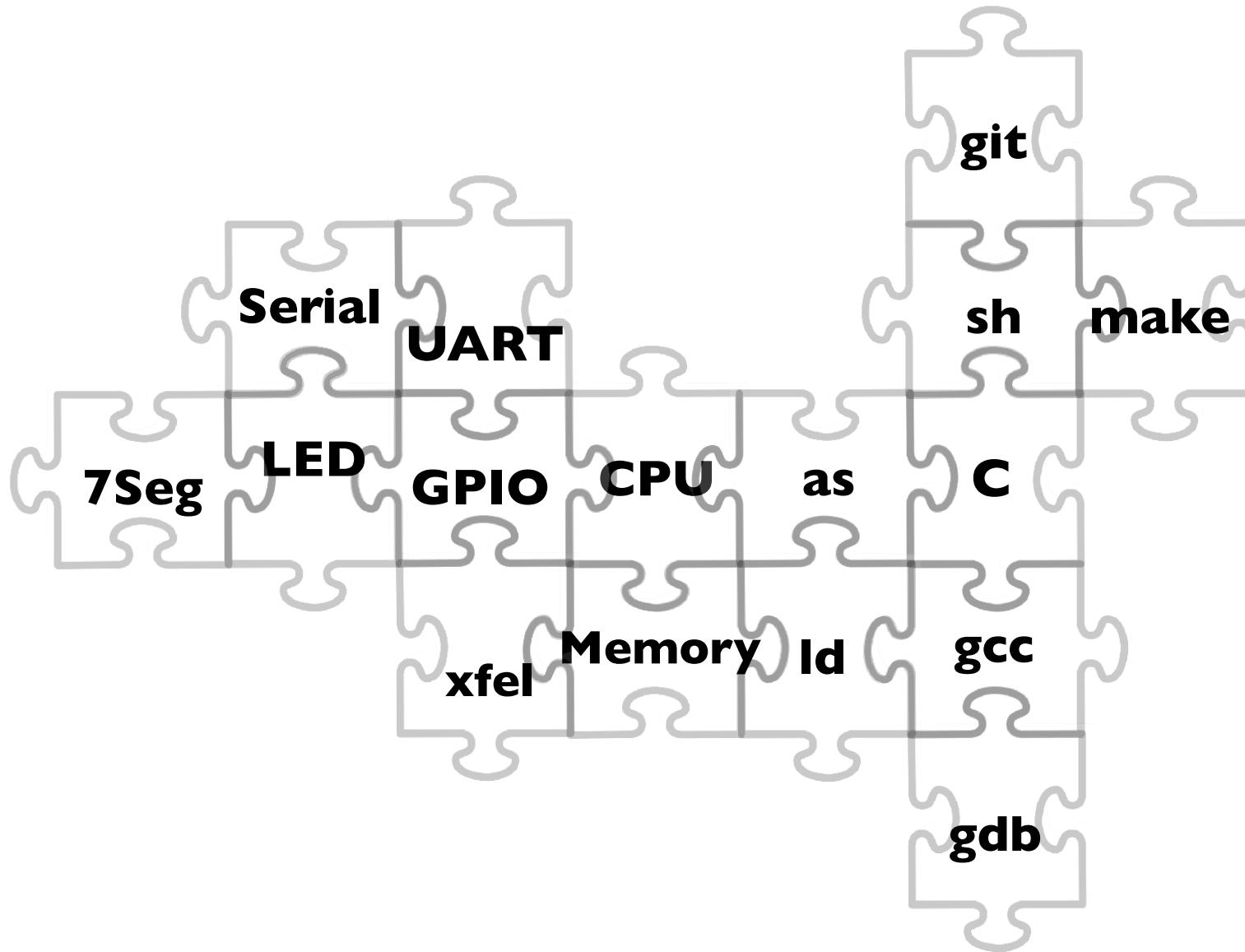












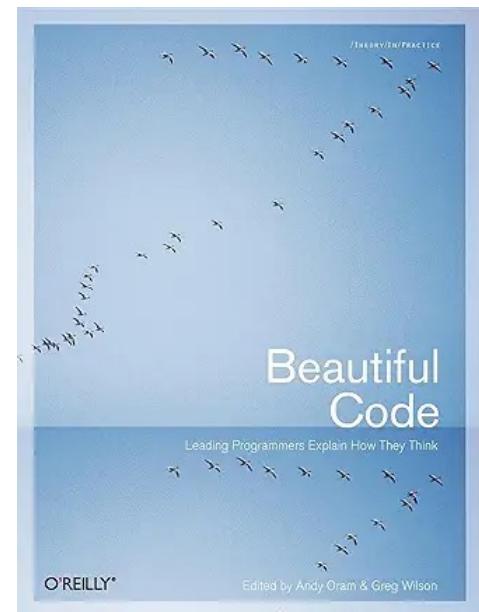
Simple and Clear Functionality is Beautiful

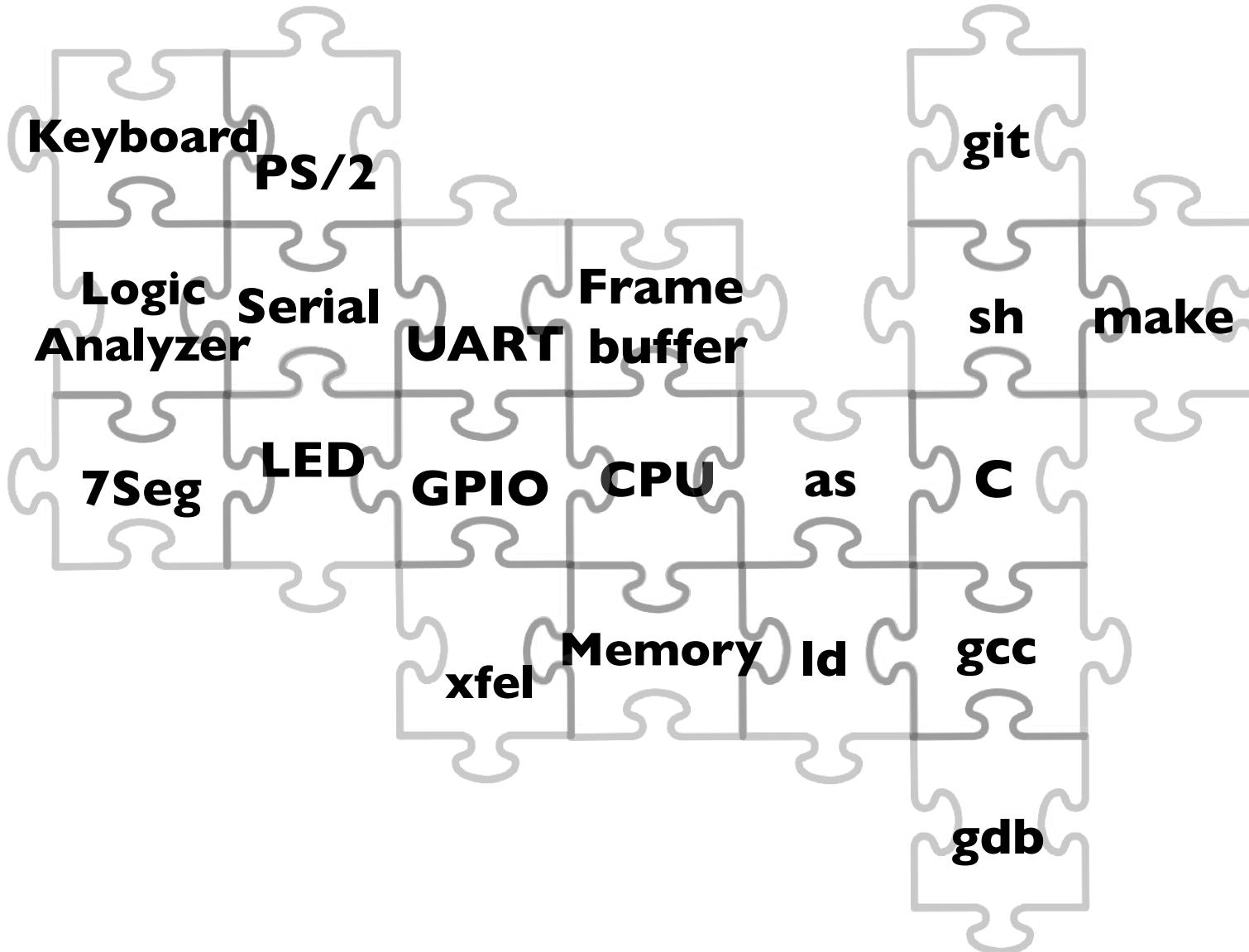
```
/* match: search for regexp anywhere in text */
int match(char *regexp, char *text) {
    if (regexp[0] == '^')
        return matchhere(regexp+1, text);
    do { /* must look even if string is empty */
        if (matchhere(regexp, text))
            return 1;
    } while (*text++ != '\0');
    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;
}

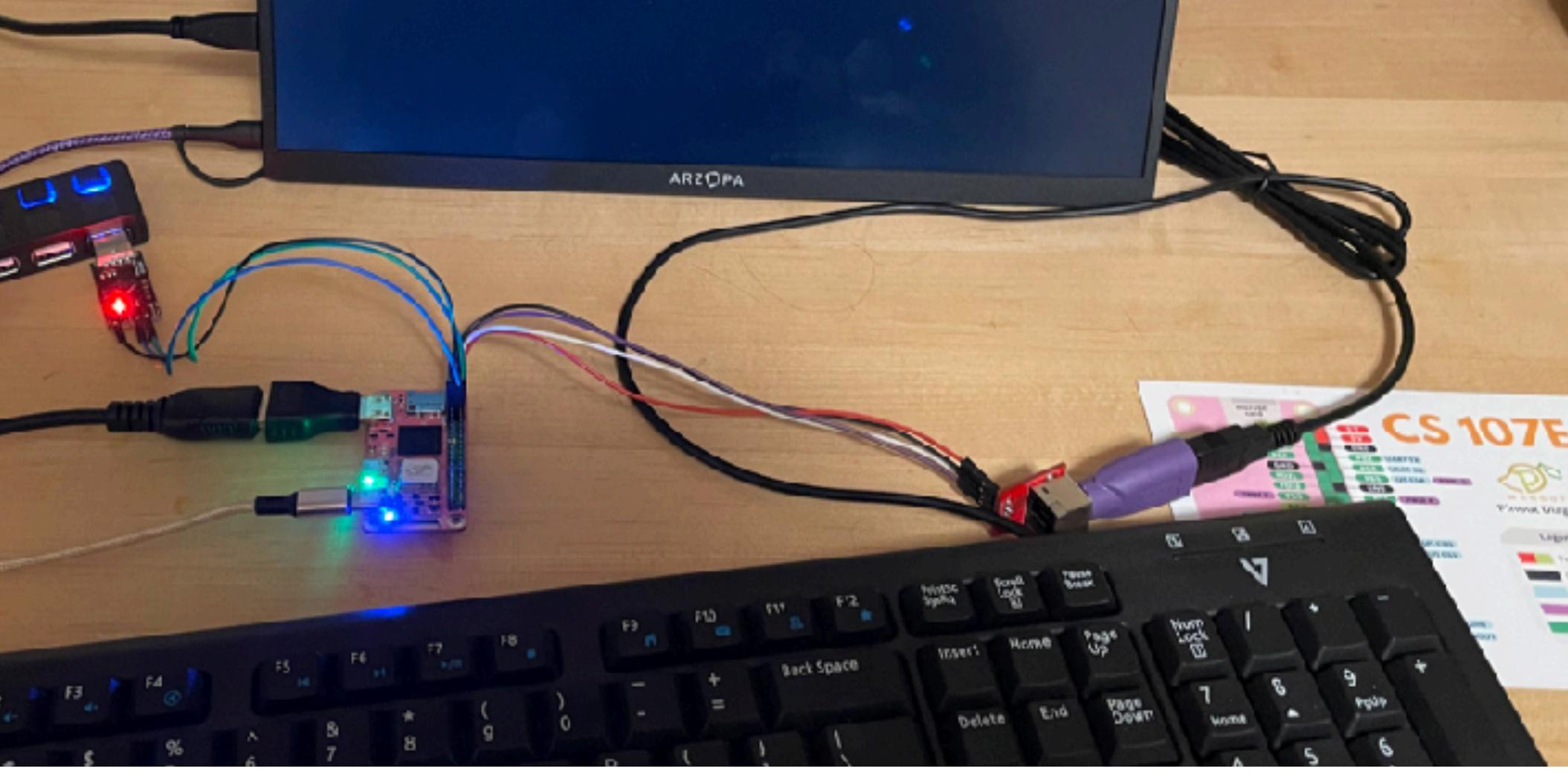
/* matchstar: search for c*regexp at beginning of text*/
int matchstar(int c, char *regexp, char *text) {
    do { /* a * matches zero or more instances */
        if (matchhere(regexp, text))
            return 1;
    } while (*text != '\0' && (*text++ == c || c == '.' ));
    return 0;
}
```

Ken Thompson regex matcher



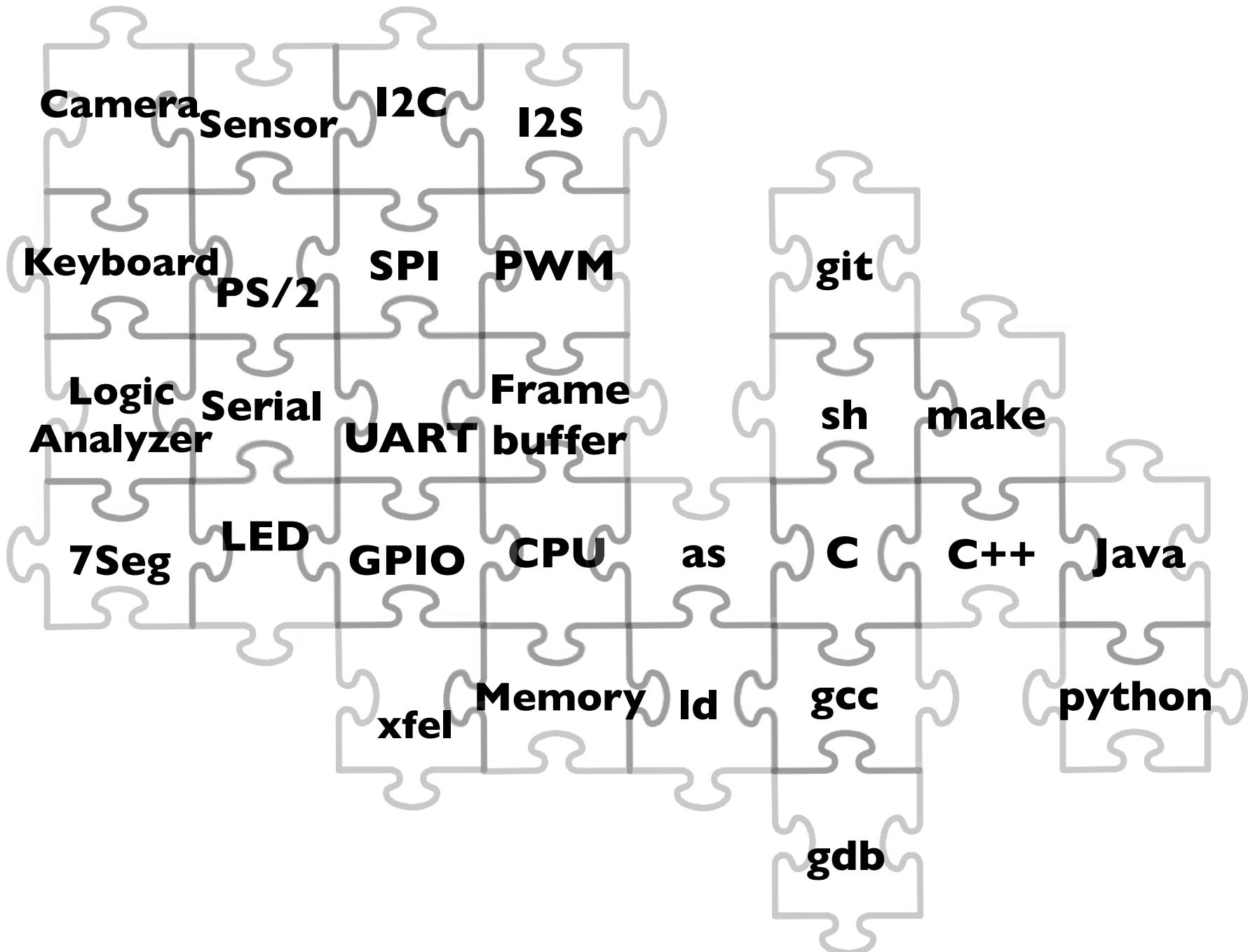


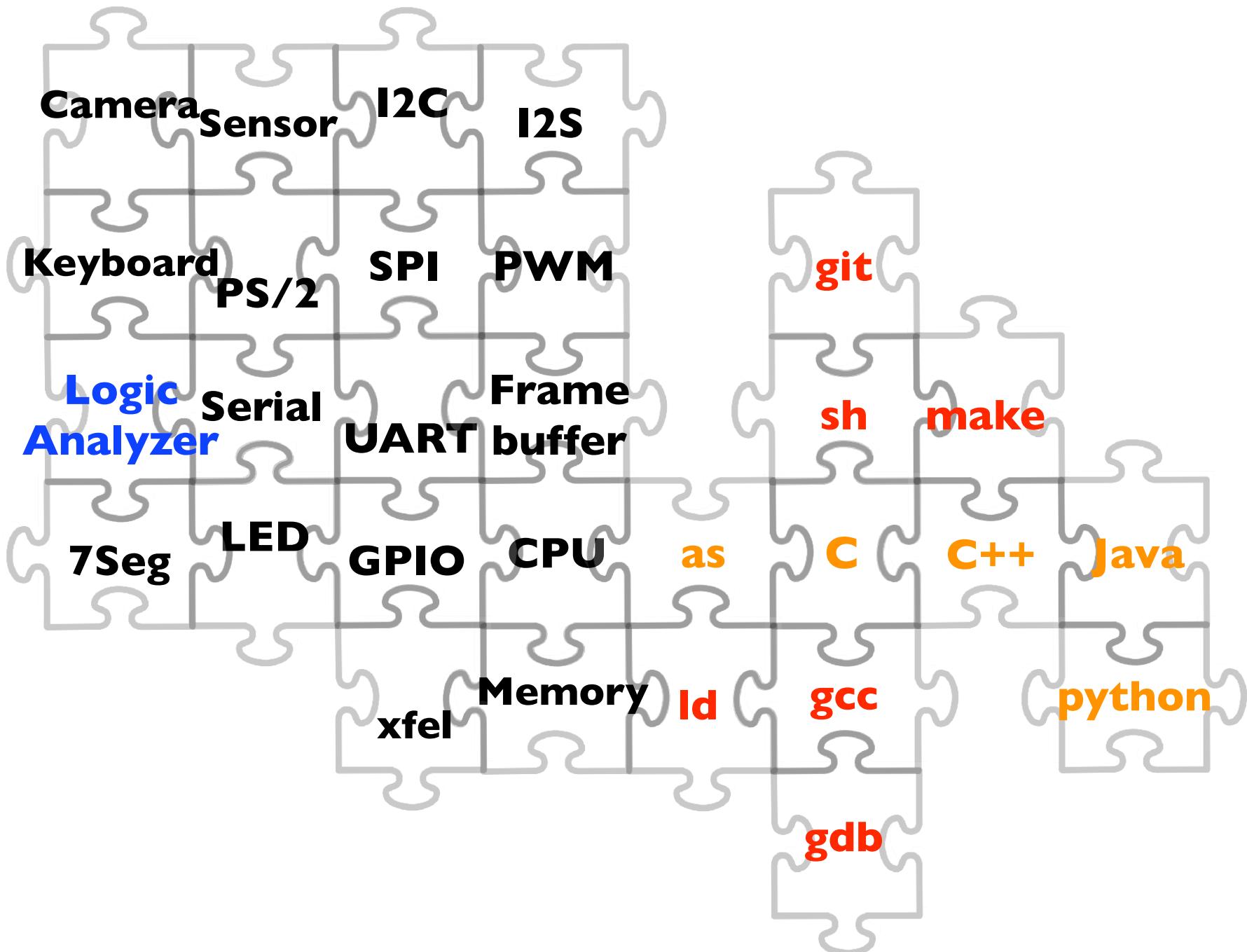
```
Welcome to the CS107E reference shell.  
Remember to type on your PS/2 keyboard!  
E23 Pi> help  
help Ecmd3          print command usage and description  
echo Eargs3          print arguments  
reboot              reboot the Rapsberry Pi  
peek Eaddr3          print contents of memory at address  
poke Eaddr3 Eval3   store value into memory at address  
history             print history of recent commands  
calc En13 Eoper3 En23 evaluate arithmetic expression  
gpio Epin3 Eaction3 control gpio pin, action is input/output/on/off  
disassemble Eaddr1Fn3 disassemble by address or function name  
E23 Pi> peek 0x20000000  
0x20000000: 00000000  
E23 Pi> gpio P010  
P010: function = 1, value = 1  
E43 Pi> echo I am Pi hear me roar  
I am Pi hear me roar  
E53 Pi>
```



The journey continues...

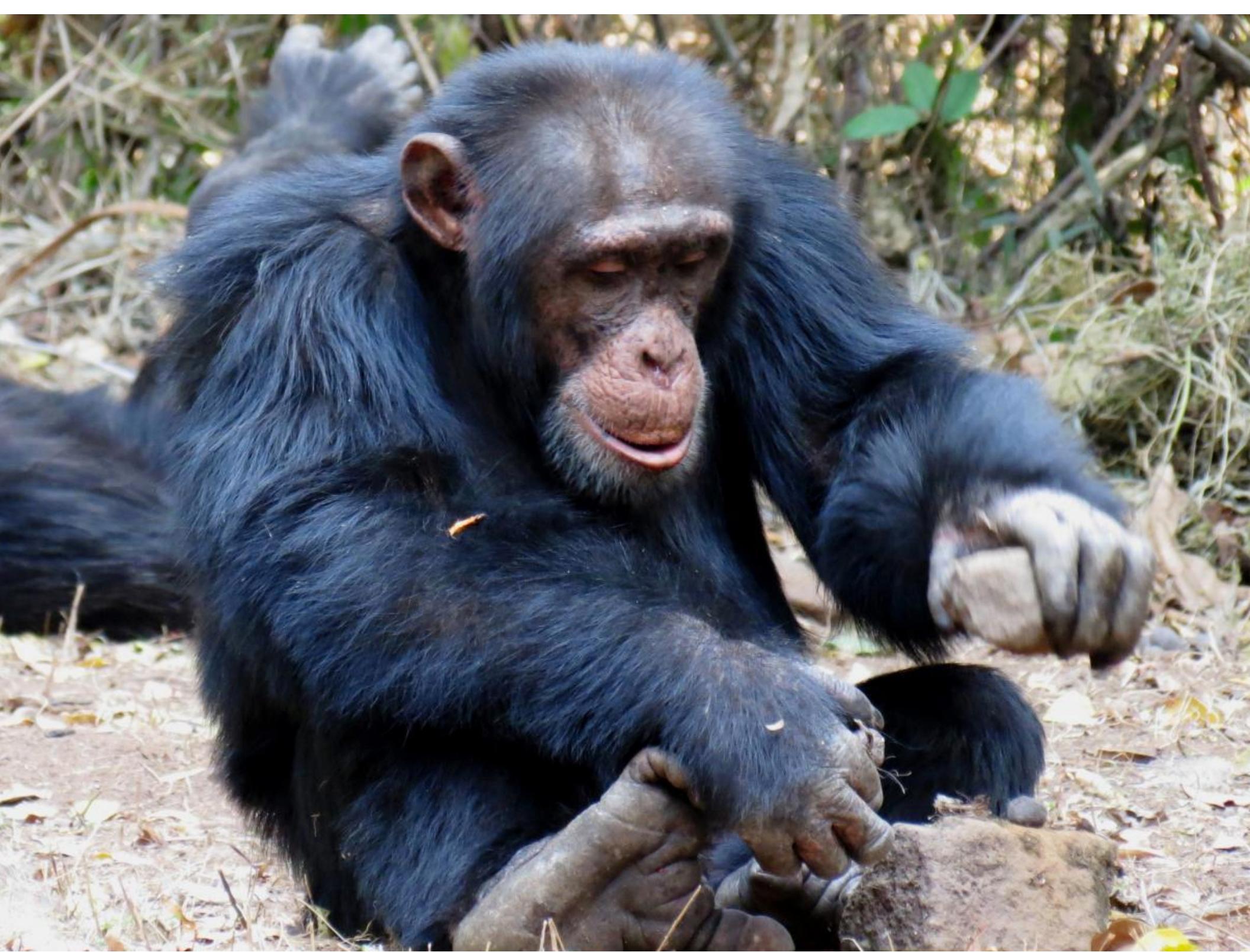






Course learning goal #2:

Master your tools





Engineering habits

Always start from a known working state, stop in a working state. If it breaks, what changed?

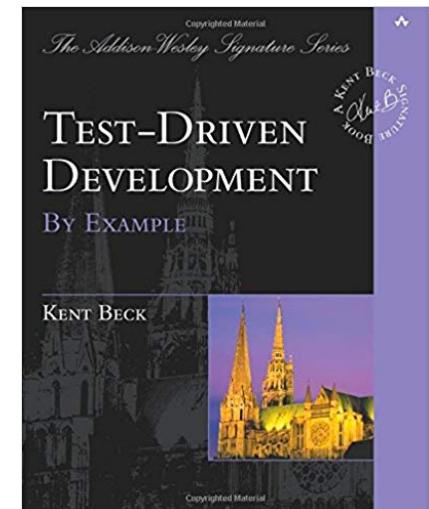
Epsilon-steps! Each step should have some small observable change. Make small edit, test, confirm, then take next step.

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

Make things visible (printf, logic analyzer, gdb)

Systematic (D&C), not random, search for bug. Form hypotheses and perform experiments. Ask: why did my code do what I observed?

Don't let bugs get you down, relish the challenge, take breaks



Be a Maker and a Doer



(A)



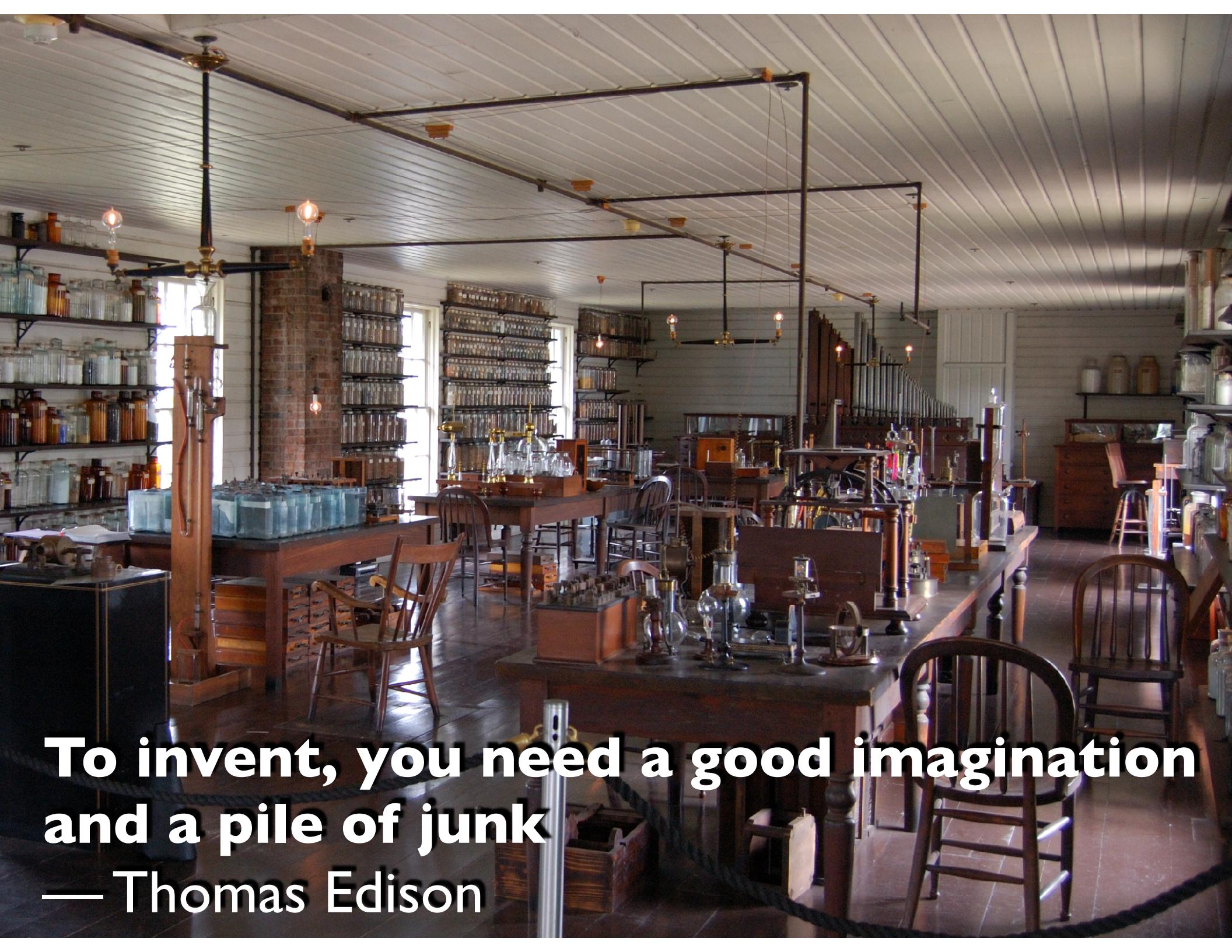
(B)



(C)

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

**Ren Ng
invents the
Light Field Camera**



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

Beyond bare metal...

Intersection of CS107 & CS107e

- Hardware/software interface
- How programs execute (asm, stack, heap)
- C compiler and linker
- Bitwise manipulation
- Computer arithmetic
- Pointers, pointers, and more pointers
- Tools (unix, git, gcc, gdb)
- Lots of coding/debugging mileage

Things unique to CS107e

- Boot sequence, bootloader
- Interacting with peripherals
- C runtime startup
- Library internals
- Interrupts, supervisor mode
- (how to do everything yourself...)

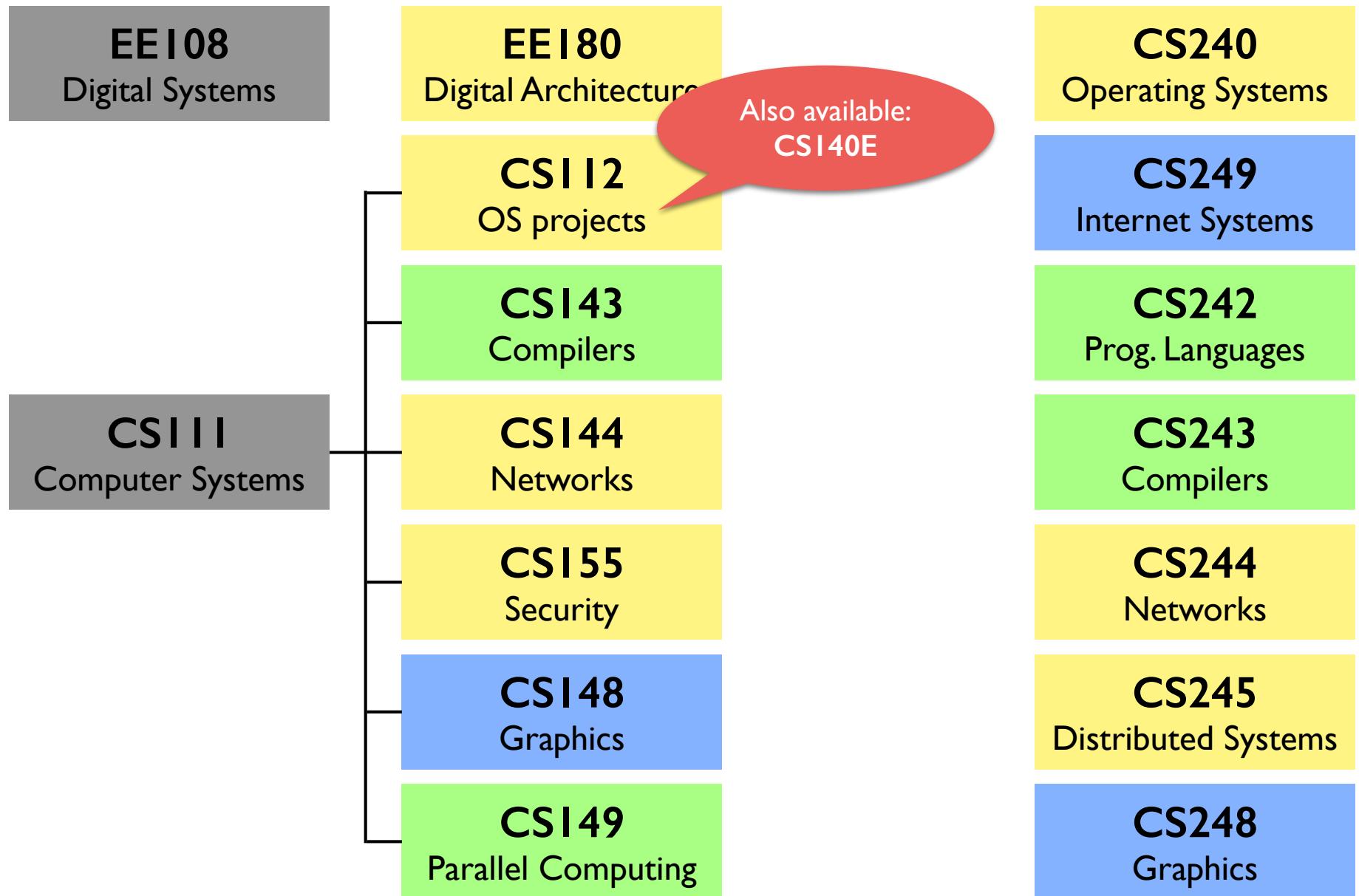
Things for you to learn

- When not to do everything yourself
- Take advantage of existing system/C libraries
- Leverage modern tools (gdb, Valgrind, gprof, ...)
- Acclimate to user mode

Opportunities!

- Section leading
- Research
- Internships
- CS for Social Good
- Co-term
- Read code and write code!
- Open source
- ...

Follow-on courses





Shoutouts!



- To our wonderful CAs and SLs
- To Apple for funding additional staff
- To all of you Mango pioneers! 

Please fill out course eval and/or share your thoughts with us