



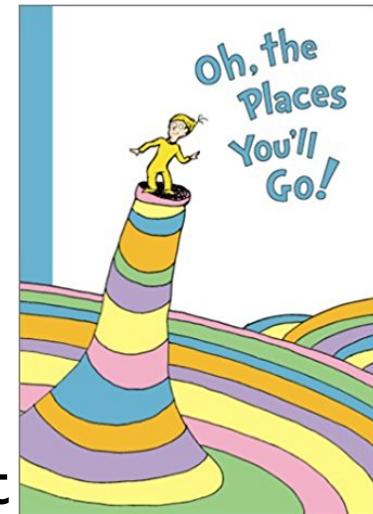
How far we've come

# CS107e Wrap Party

Where we started



Where to go next



# **Lab this week is project check-in**

Progress/milestones, path to completion

## **Project demo day**

Fri Mar 22 9-11:30am in B21

Each team gives 2-minute pitch/demo to audience

Then circulate and check out each other's projects

## **Project submission**

Due 9am Fri Mar 22, final GitHub commit of code

Include README.md (description, pictures, attributions)

## **End-quarter logistics**

Return keyboards and mice

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

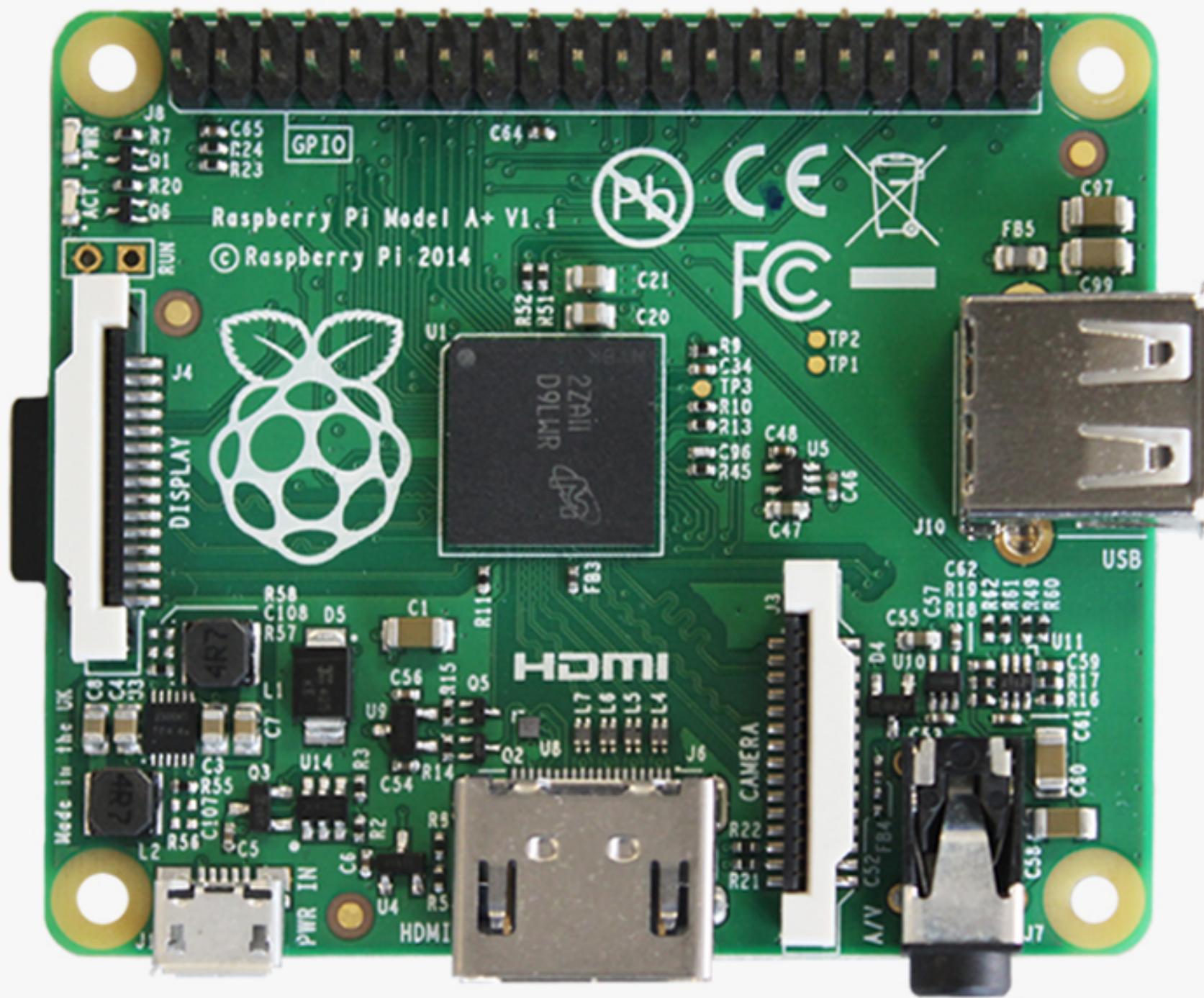
Submit receipts for reimbursements

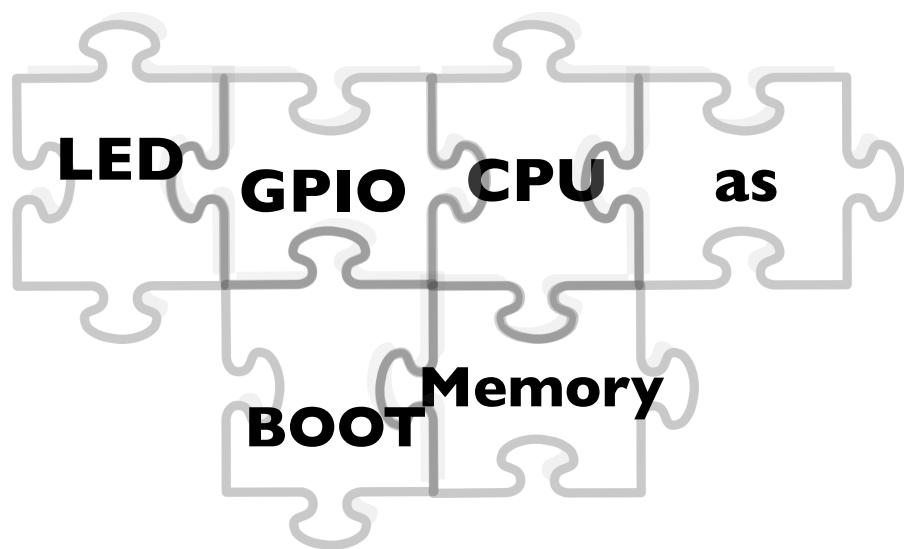
Please fill out course evaluation, offer feedback for future offerings

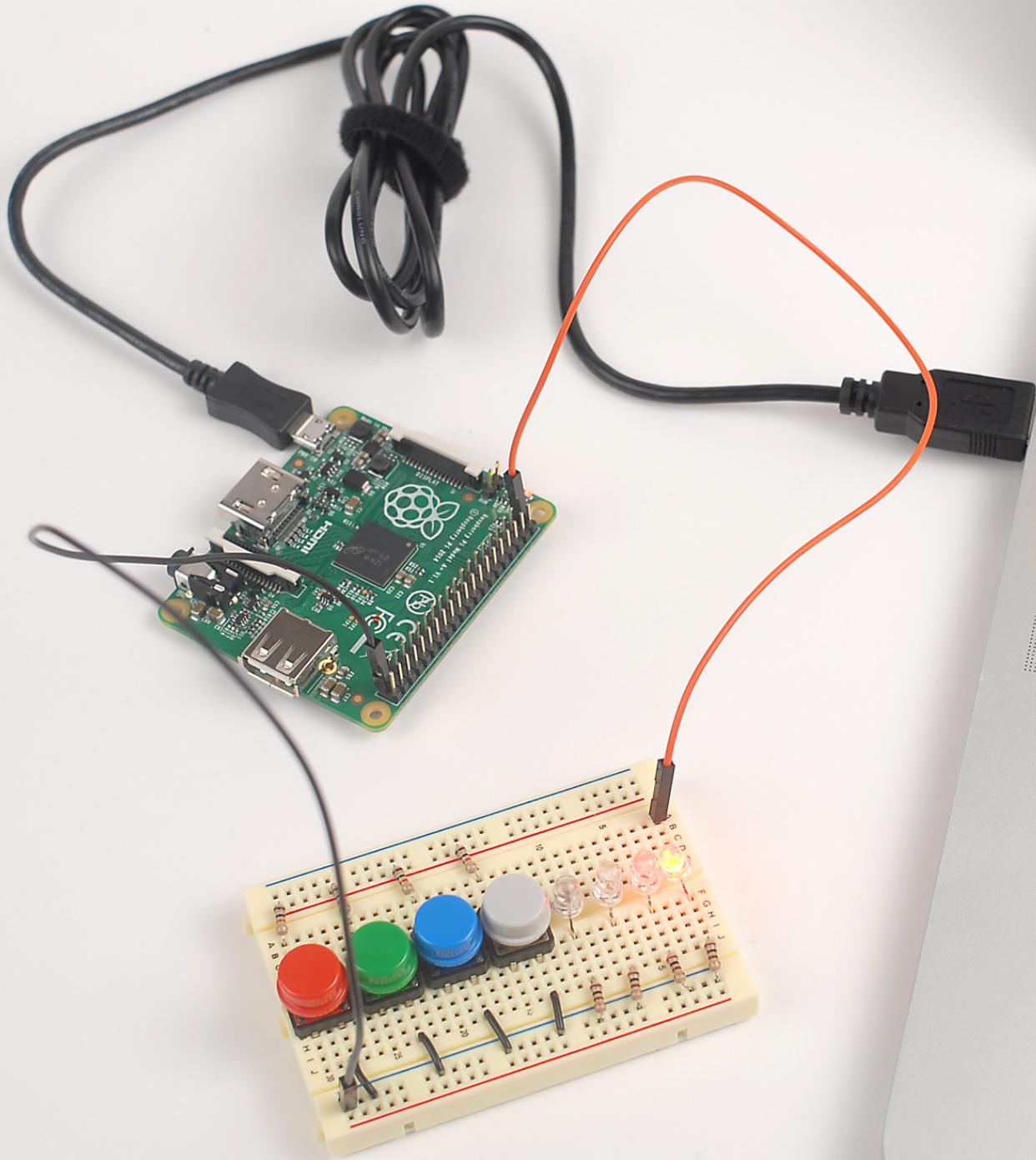
Pi kit is yours to keep!

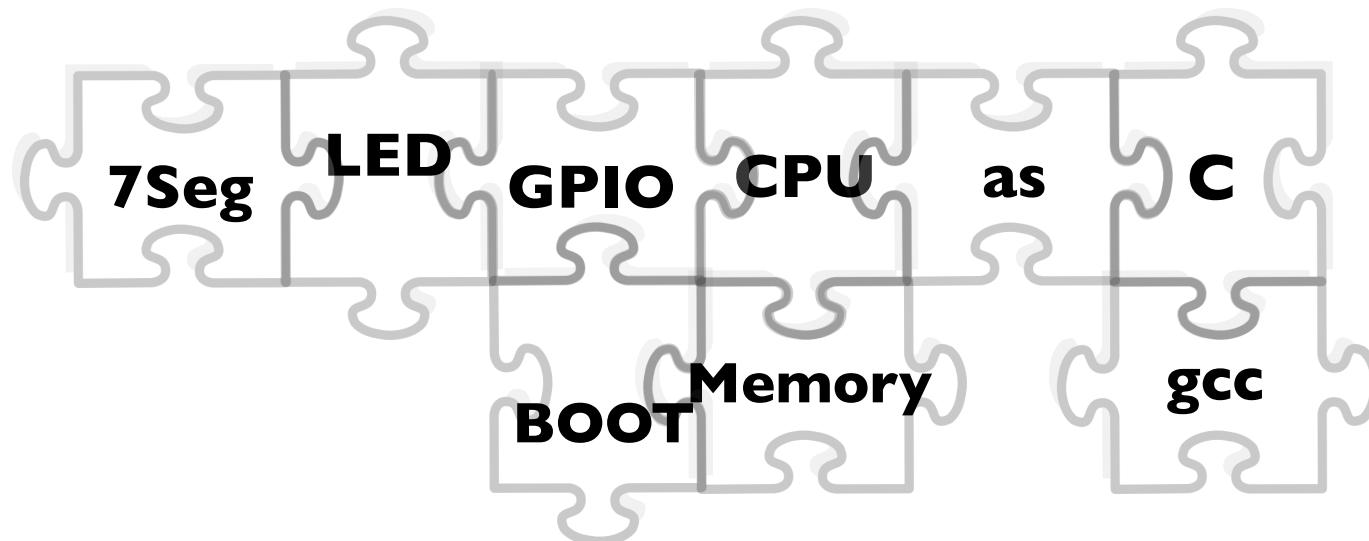
Course learning goal #1:

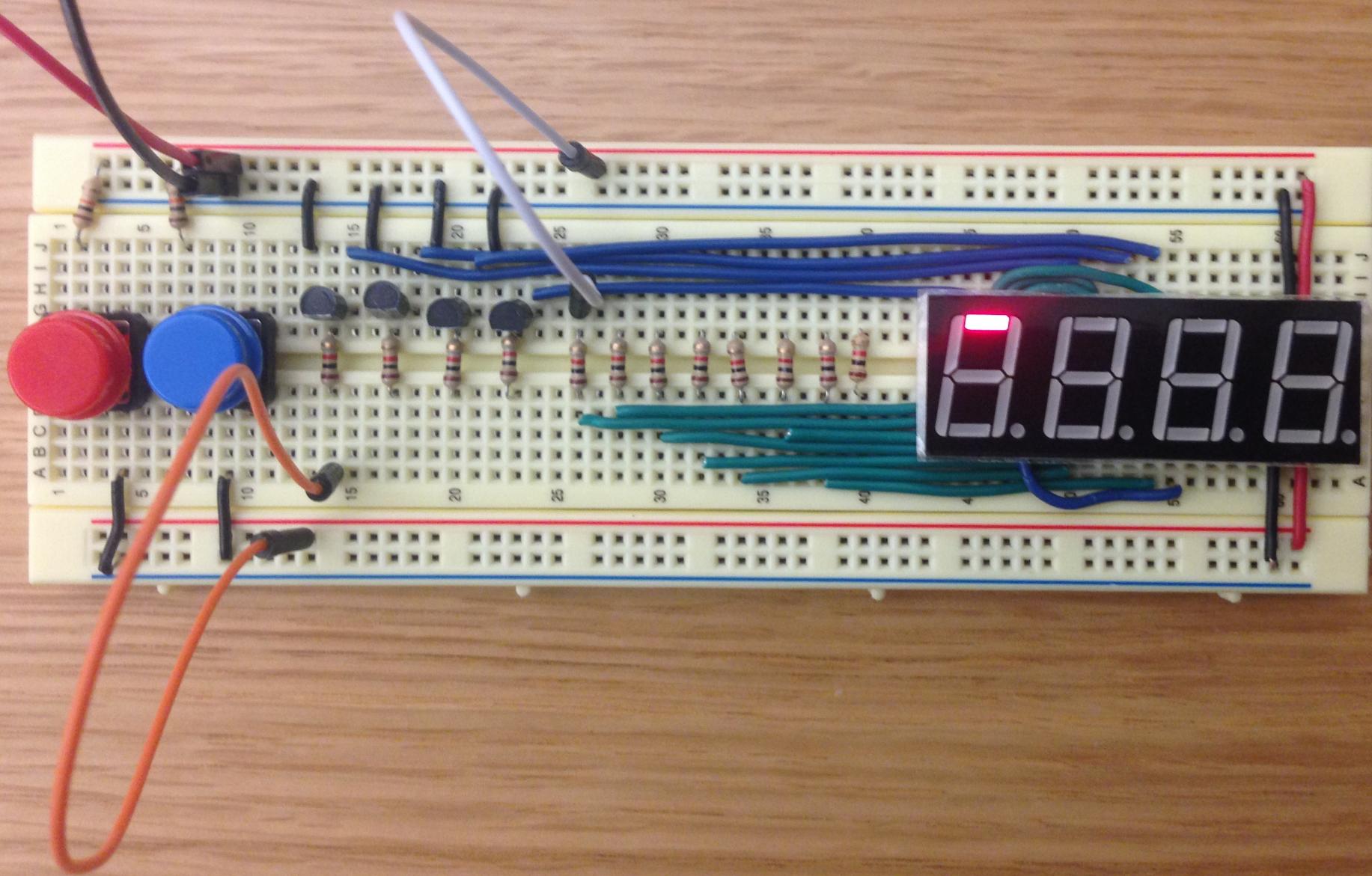
**Understand how  
computers work**

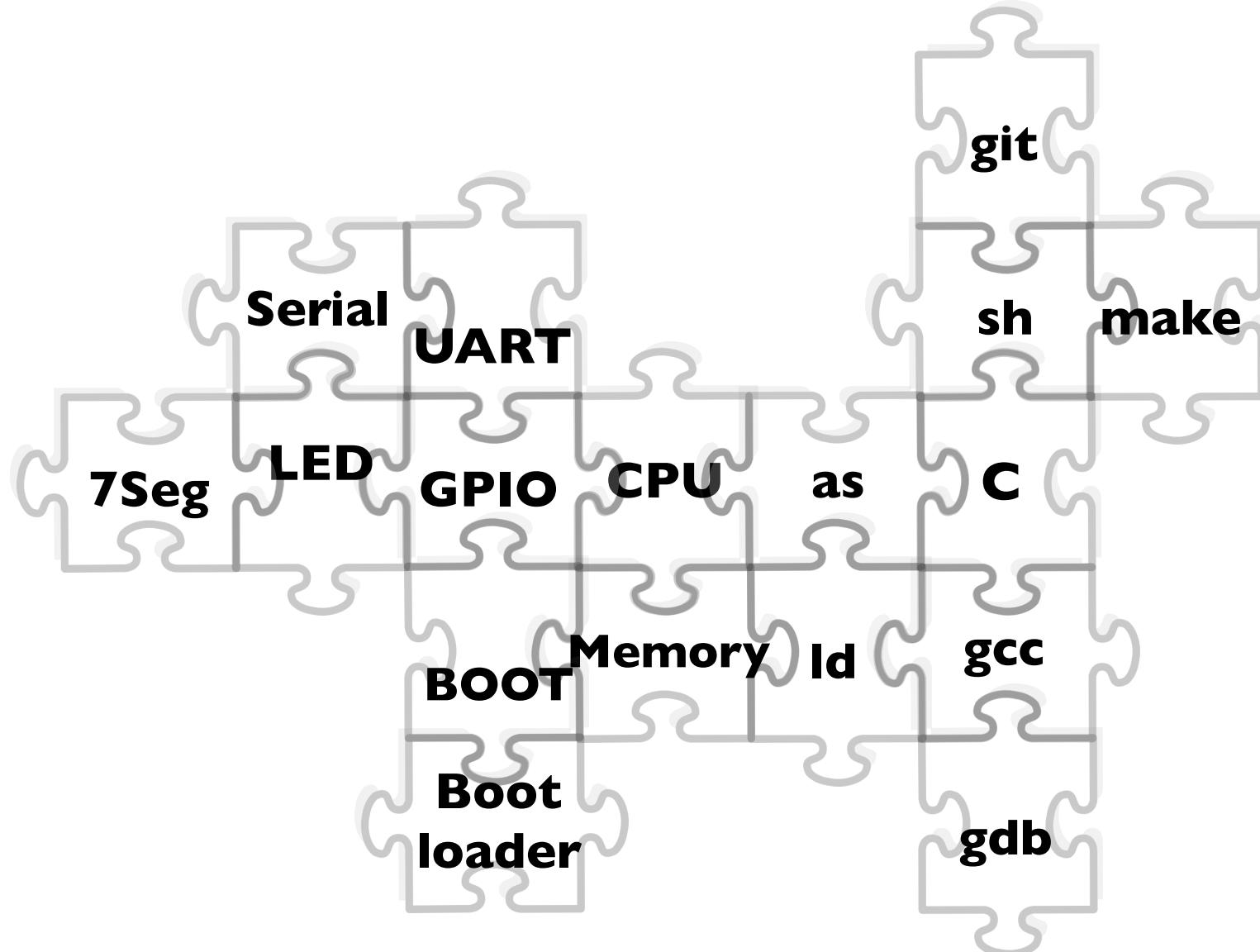


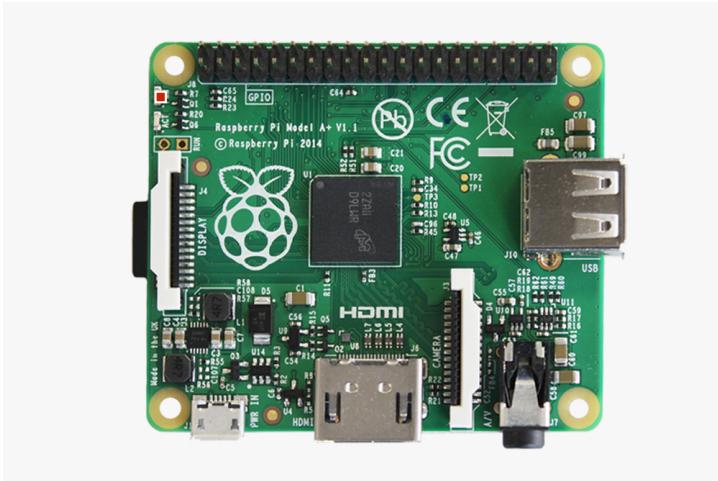




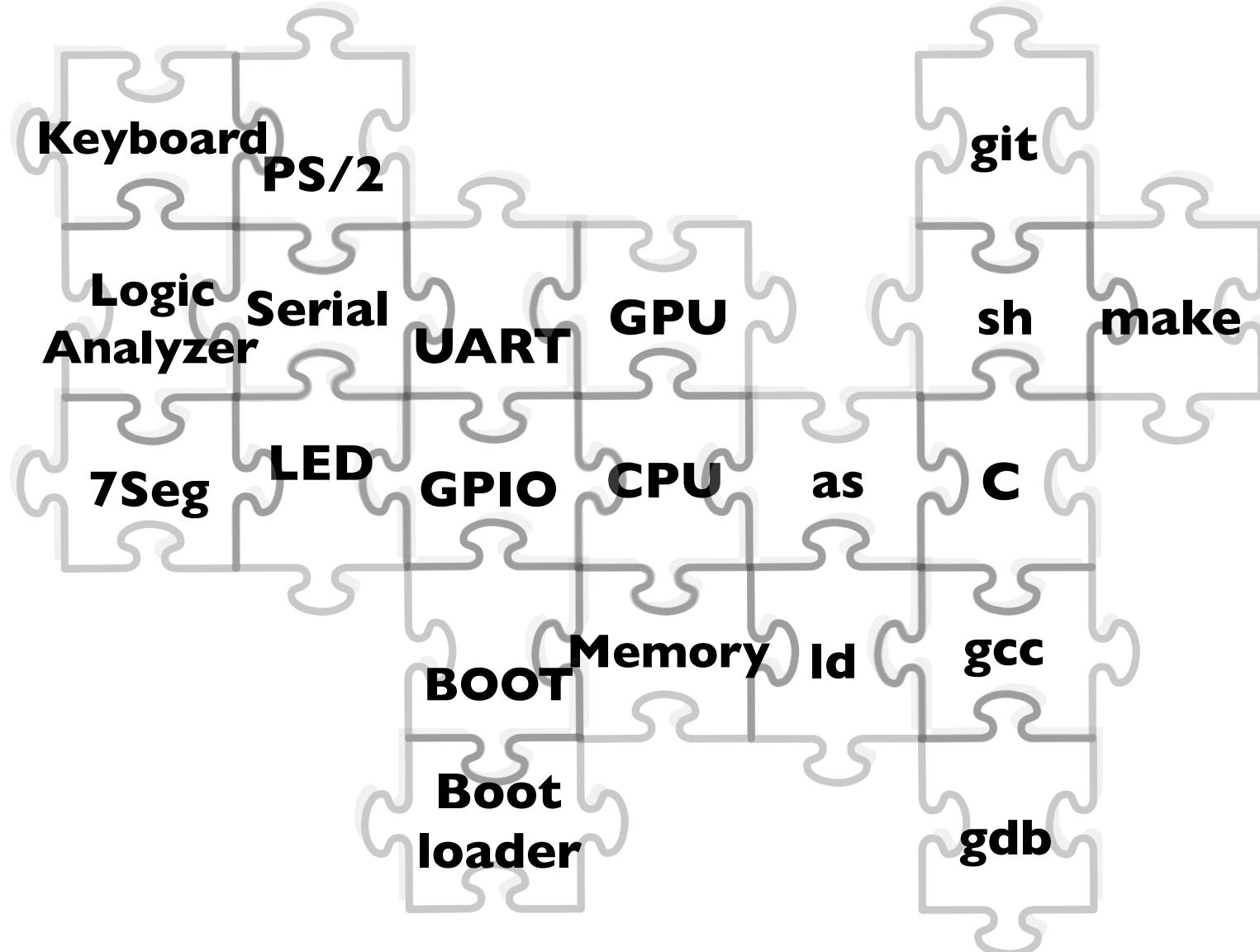




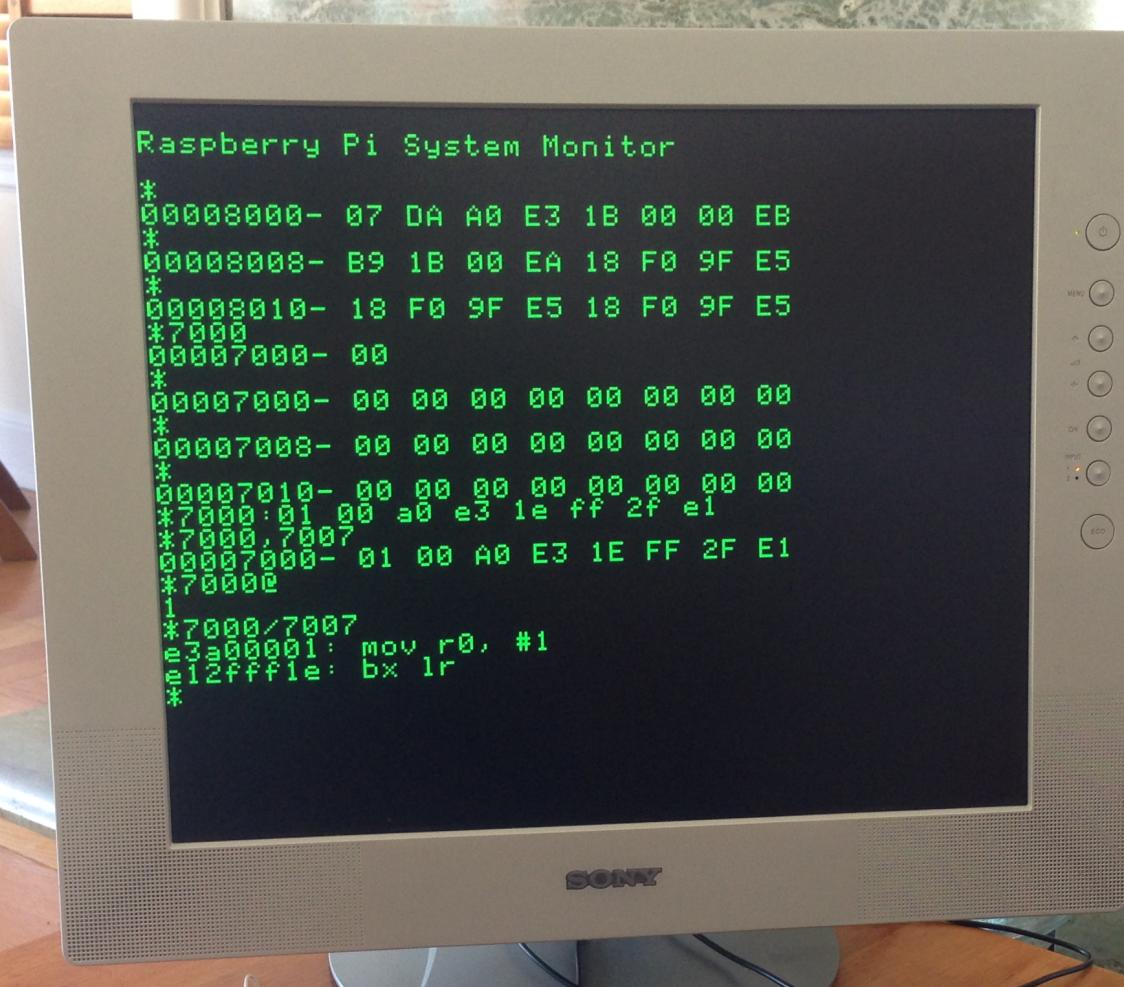
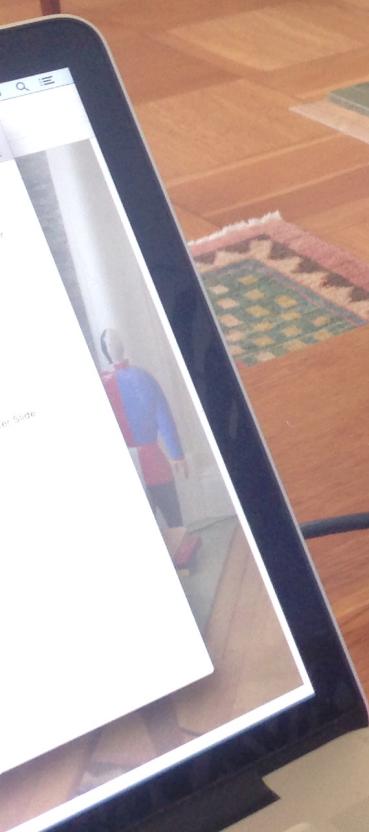




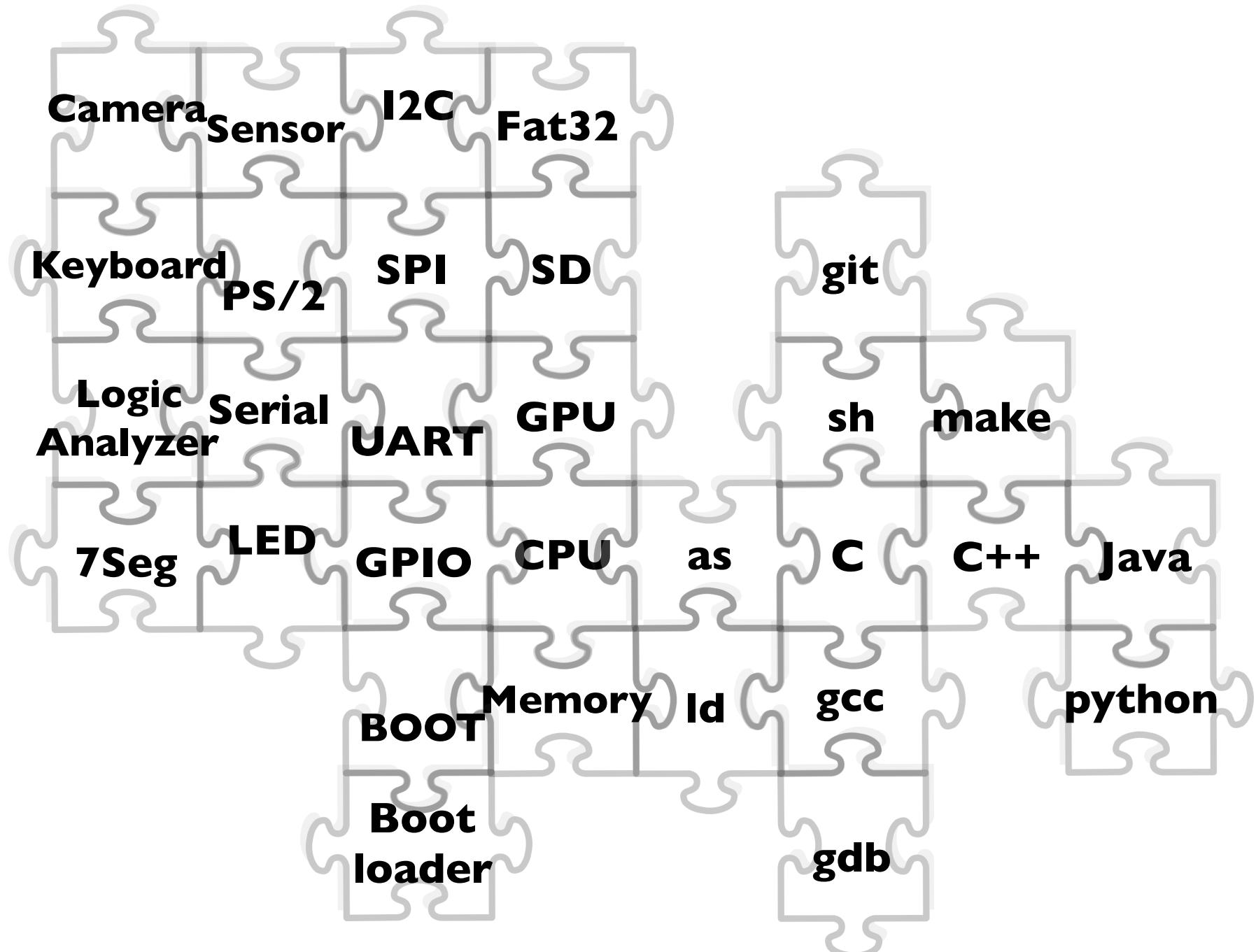
**File main.c, line 107: Assertion 'ptr != NULL' failed.**

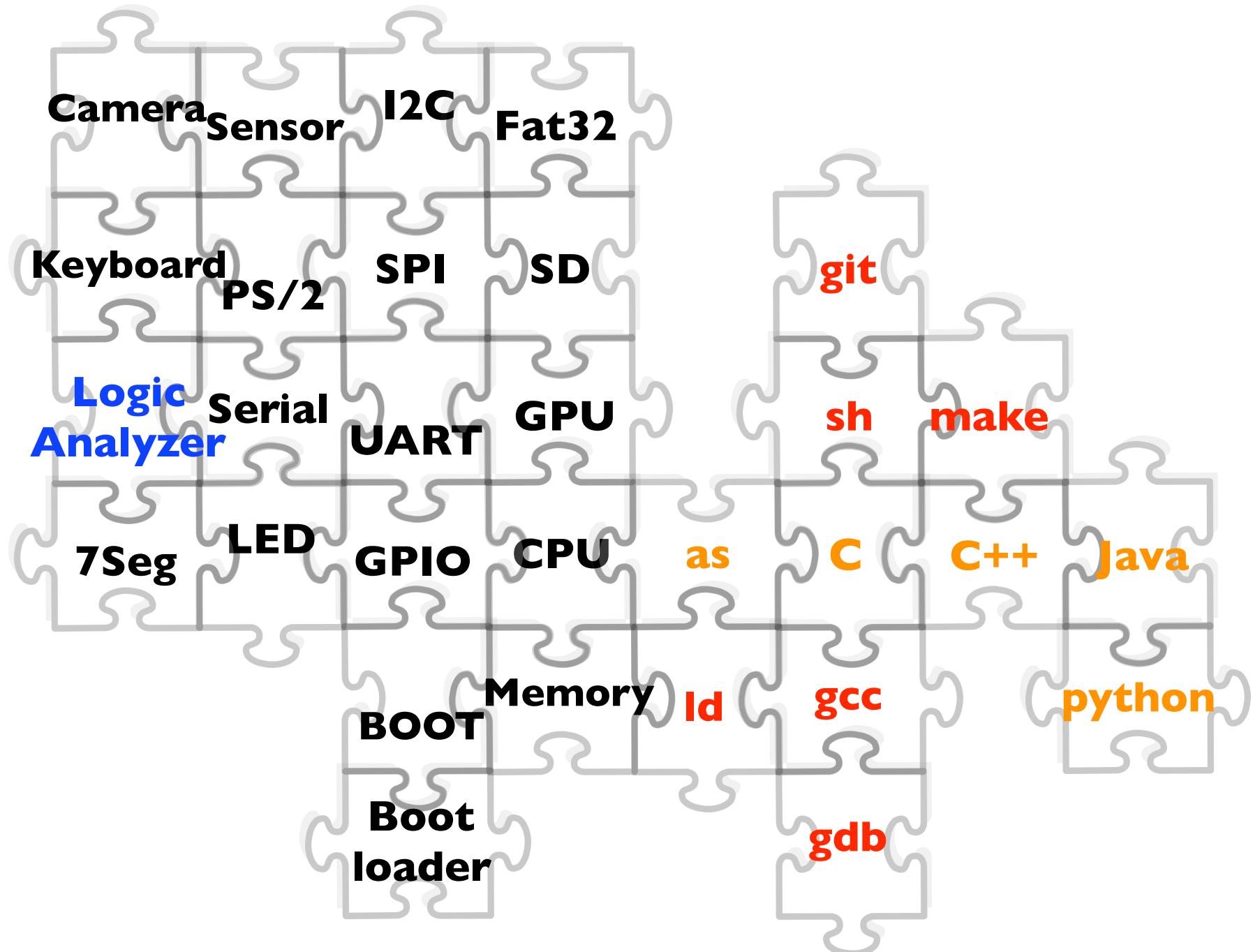


```
Raspberry Pi System Monitor  
*  
00008000- 07 DA A0 E3 1B 00 00 EB  
*  
00008008- B9 1B 00 EA 18 F0 9F E5  
*  
00008010- 18 F0 9F E5 18 F0 9F E5  
*7000  
00007000- 00  
*  
00007000- 00 00 00 00 00 00 00 00  
*  
00007008- 00 00 00 00 00 00 00 00  
*  
00007010- 00 00 00 00 00 00 00 00  
*7000:01 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  
**
```



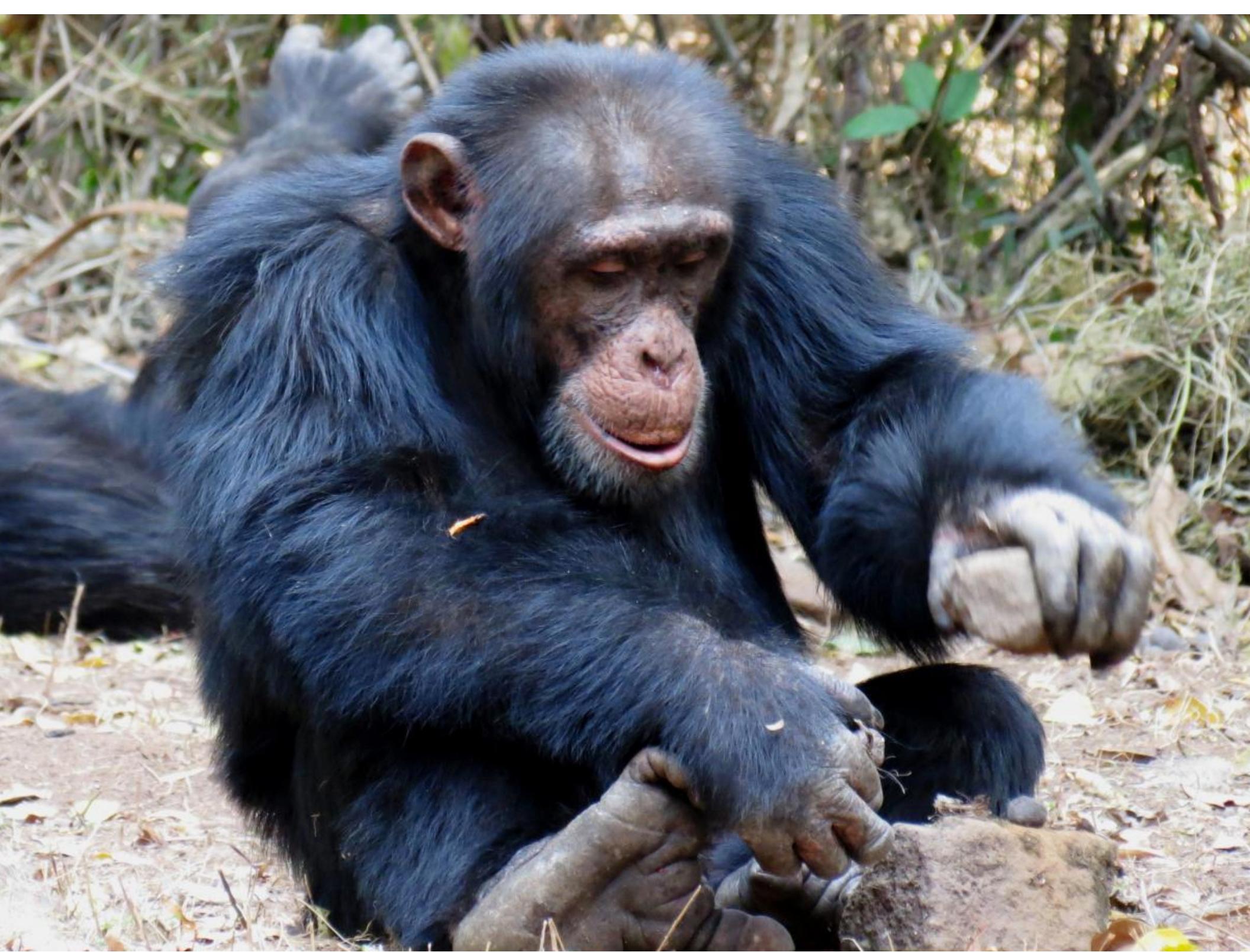
**It keeps growing!**





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! Make simple small change, test carefully, then take another small step. Programming is much simpler if you figure out a good order to write the code!

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 be frustrated by bugs, relish the challenge, take frequent breaks



**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 CSI07E

- 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

