

CS107E

Computer Systems from the Ground Up

Christos Kozyrakis

Maria Fernandez, Matt Trost

Liana Keesing, Anna Mistelle

Spring 2022

<https://cs107e.github.io/>

Have you ever wondered ...

- how a computer represents data?
- what operations a computer understands?
- how a program executes?
- what happens when a user types on keyboard?
- how text and drawing appears on a display?
- how things *really* work inside that wondrous box?

Studying and understanding **computer systems** will answer these questions

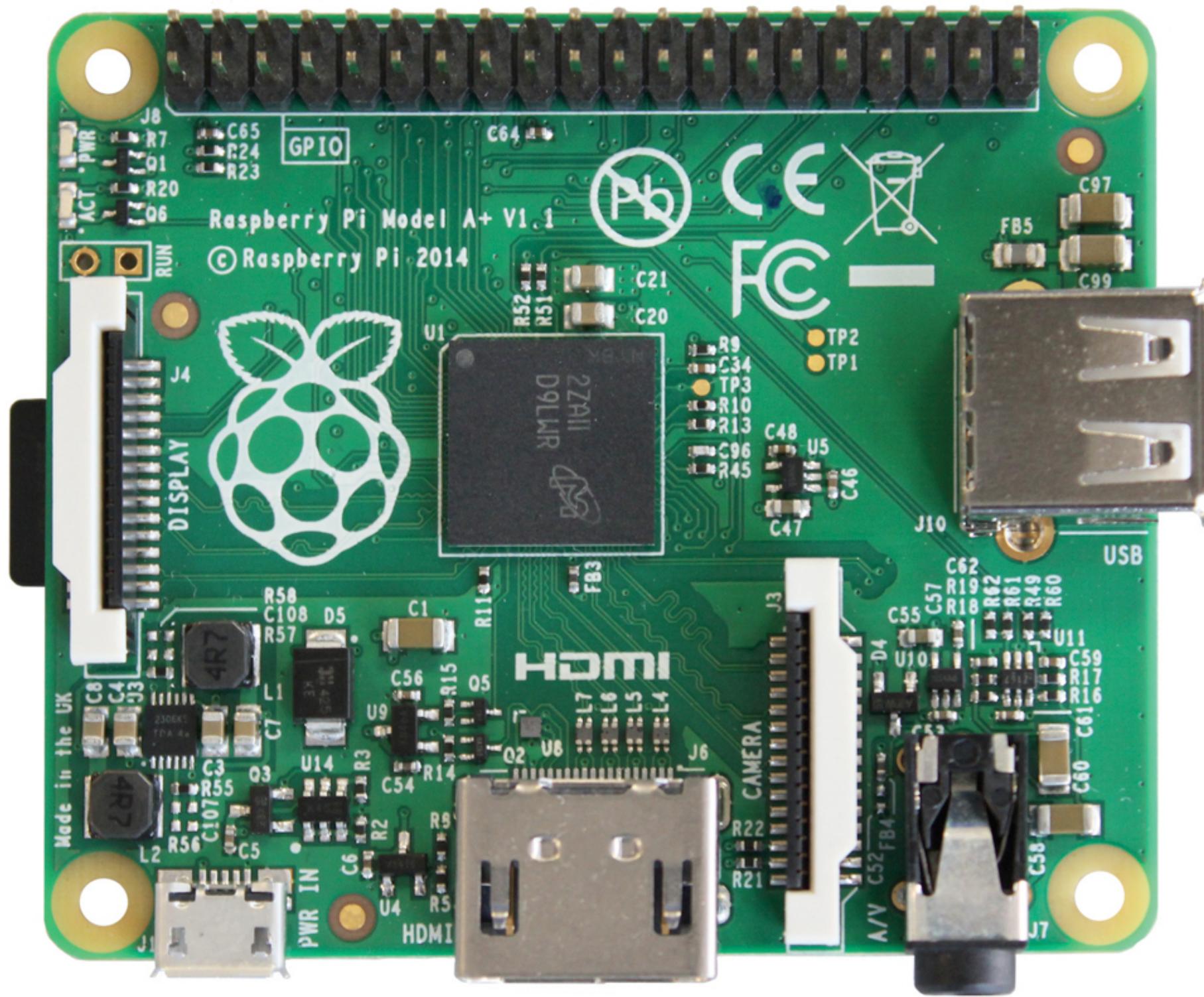
What system to study and where to start?

Bare Metal on the Raspberry Pi

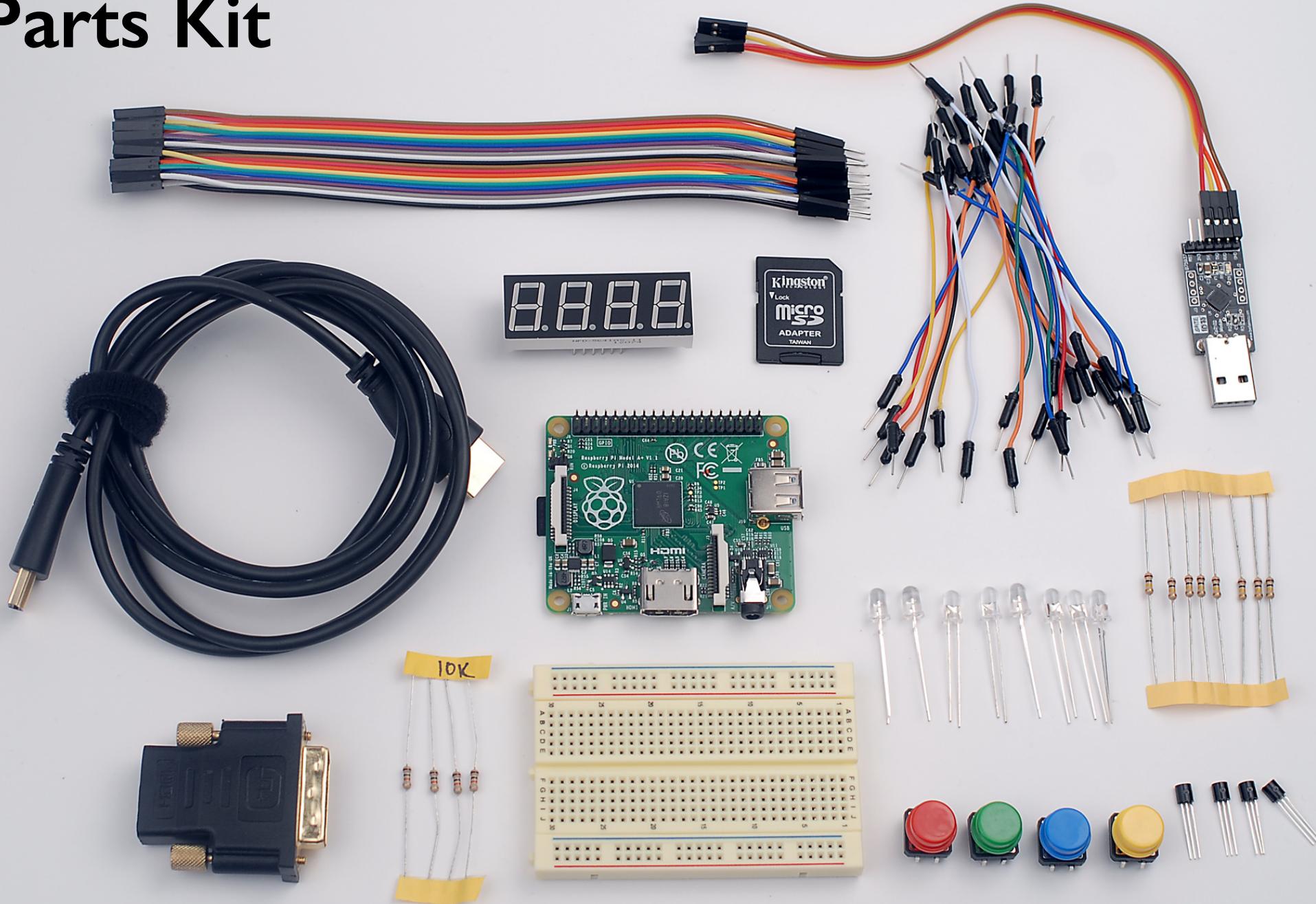
Definition: **Bare metal** programming involves no operating system (programmer constructs libraries)

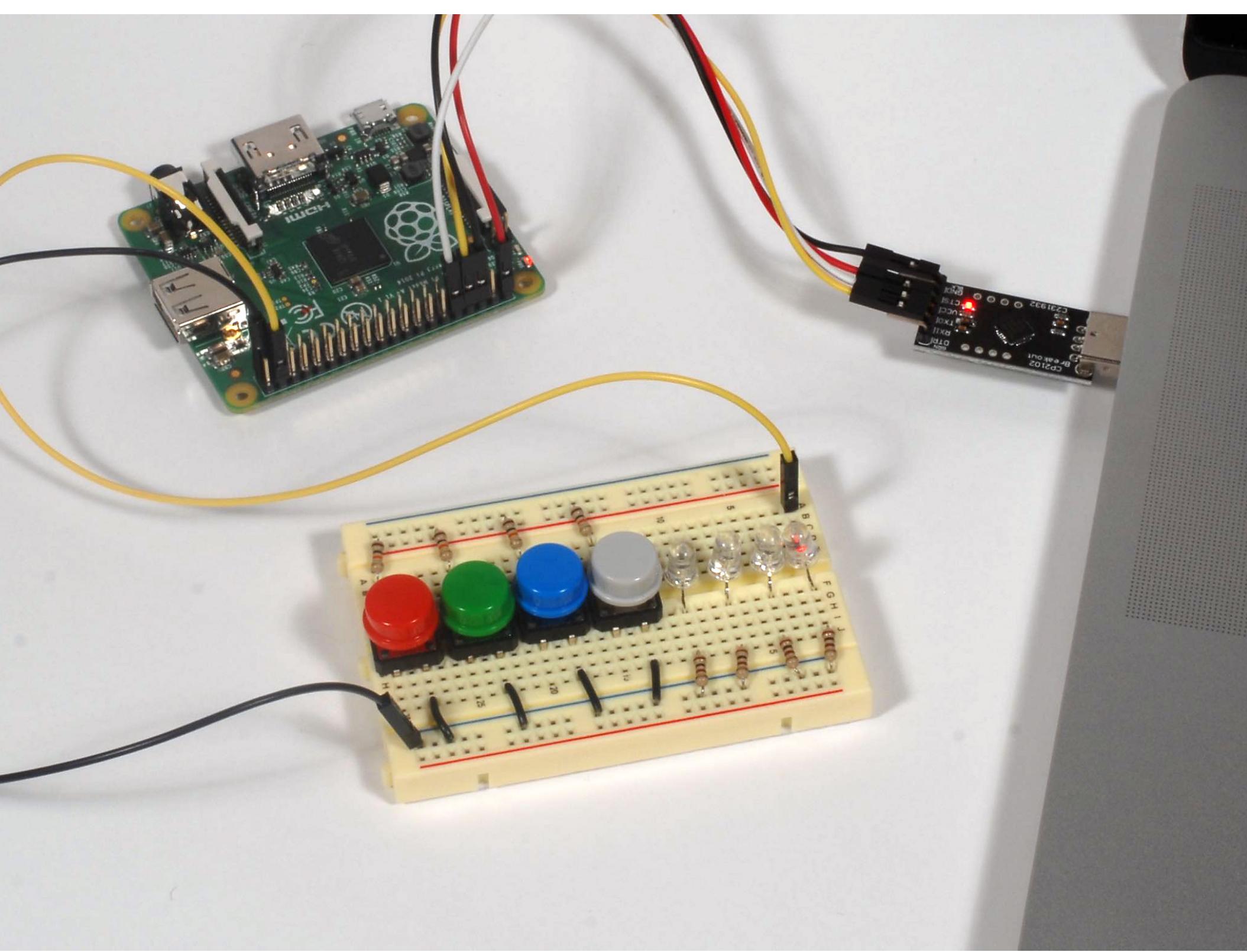
Bare metal programs boot and startup on their own, and directly control peripherals

Understanding the foundations makes all of the layers above much simpler



Parts Kit





Raspberry Pi Shell

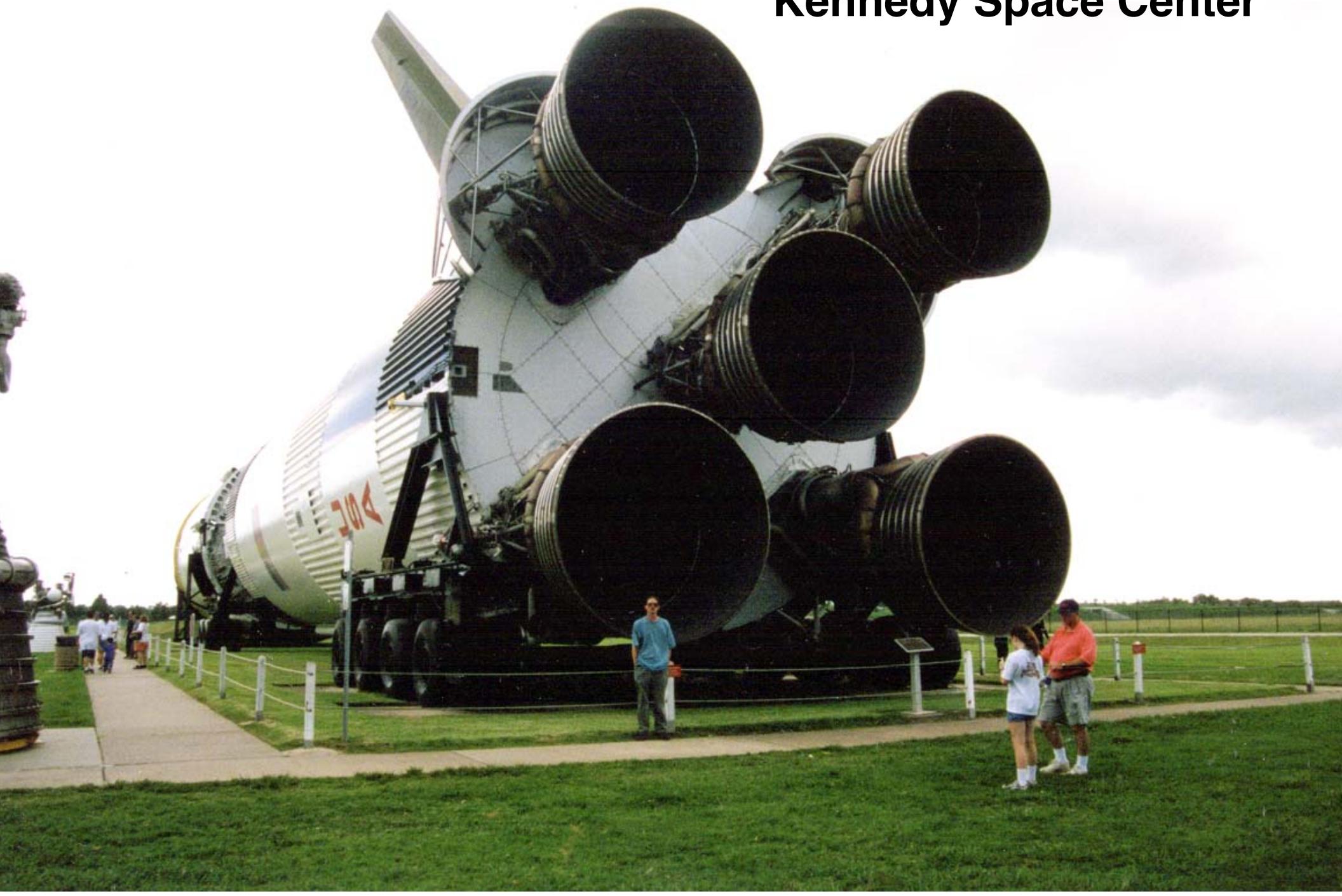
```
Raspberry Pi System Monitor  
*  
*00008000- 07 DA A0 E3 1B 00 00 EB  
*00008008- B9 1B 00 EA 18 F0 9F E5  
*00008010- 18 F0 9F E5 18 F0 9F E5  
*7000  
00007000- 00  
*00007000- 00 00 00 00 00 00 00 00  
*00007008- 00 00 00 00 00 00 00 00  
*00007010- 00 00 00 00 00 00 00 00  
*7000:01 00 a0 e3 1e ff 2f e1  
*7000,7007  
00007000- 01 00 A0 E3 1E FF 2F E1  
*70000  
1  
*7000/7007  
e3a00001: mov r0, #1  
e12fffffe: bx lr  
*
```

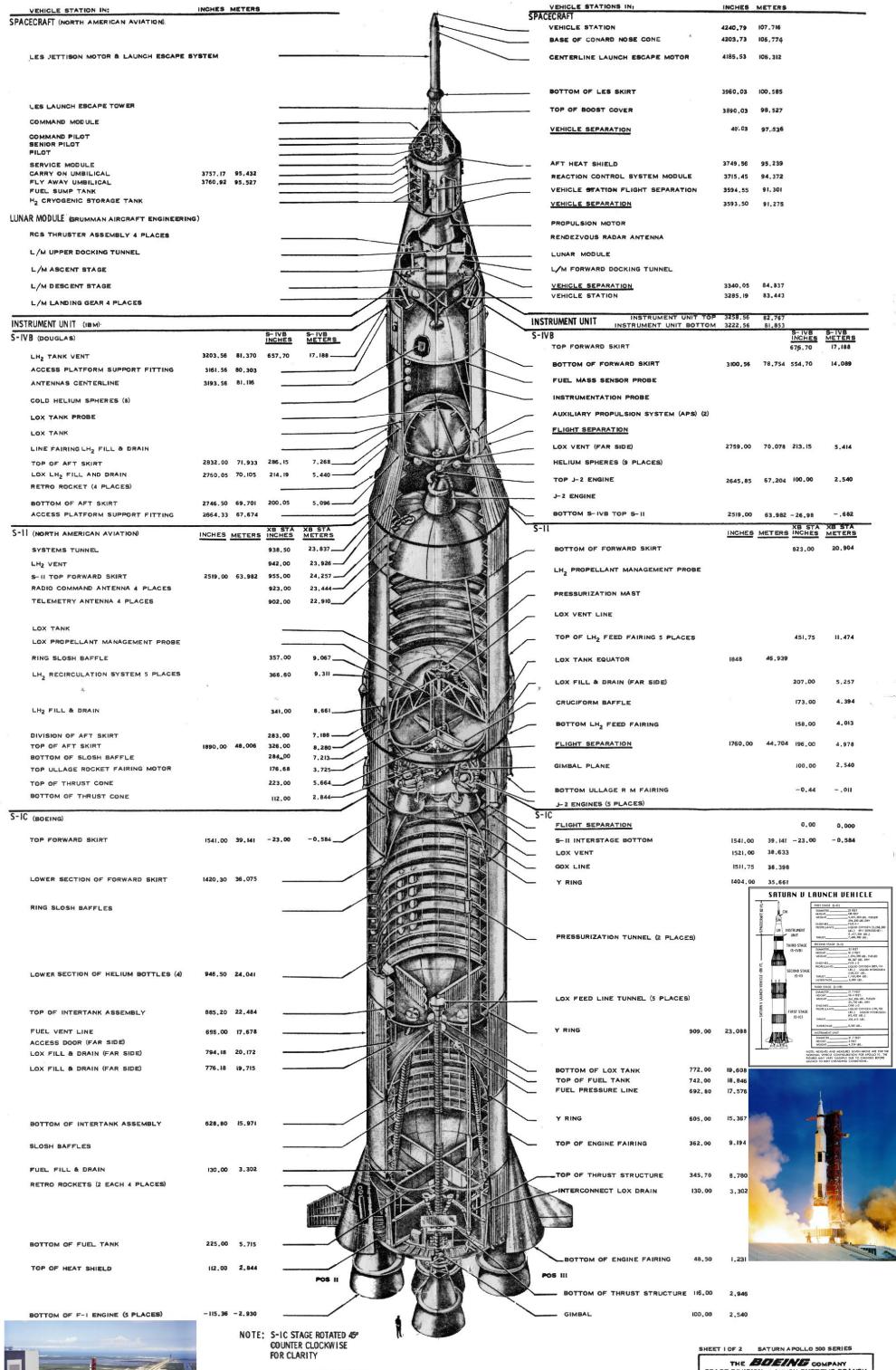


Learning Goal 1

**Understand how computers
represent data and instructions,
execute programs,
and control peripherals**

Saturn V Kennedy Space Center

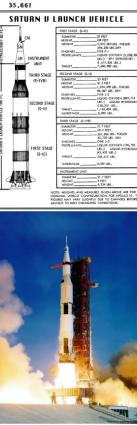




Command Module 64,000 lbs

Saturn V 6,200,000 lbs

Payload 1.5% of total weight



SHEET 1 OF 2 SATURN APOLLO 300 SERIES

THE BOEING COMPANY
SPACE DIVISION HUNTSVILLE BRANCH
HUNTSVILLE, ALA. 35807

SATURN V APOLLO
FLIGHT CONFIGURATION

DRAWN BY: [Signature] **DATE:** 1 JANUARY 1967
DRAFTER: [Signature] **GRADE:** 1
ENGINEERED BY: [Signature] **GRADE:** 1
HUNTSVILLE
ENGINEERING

Falcon 9





Understanding is Empowering

Understand ...

ARM processor and memory architecture

Peripherals: GPIO, timers, UART, ...

Assembly language and machine code

Low-level representation of information / bits

From assembly language to C

Function calls and stack frames

Serial communication and strings

Modules and libraries: Building and linking

Memory management: Memory map & heap

Learning Goal 2

Master your tools

Software Tools

UNIX command line: bash, cd, ls, ...

Text editor: vim, emacs, sublime, ...

Programming languages: C, ...

Compiler: gcc

Assembler: as

Linker/loader: ld

binutils: nm, objcopy, objdump, ...

make

git and github.com

documentation: markdown

Software Tools

UNIX command line: bash, cd, ls, ...

Text editor: vim, emacs, sublime, ...

Programming languages: C, ...

Compiler: gcc

Assembler: as

Linker/loader: ld

binutils: nm, objcopy, objdump, ...

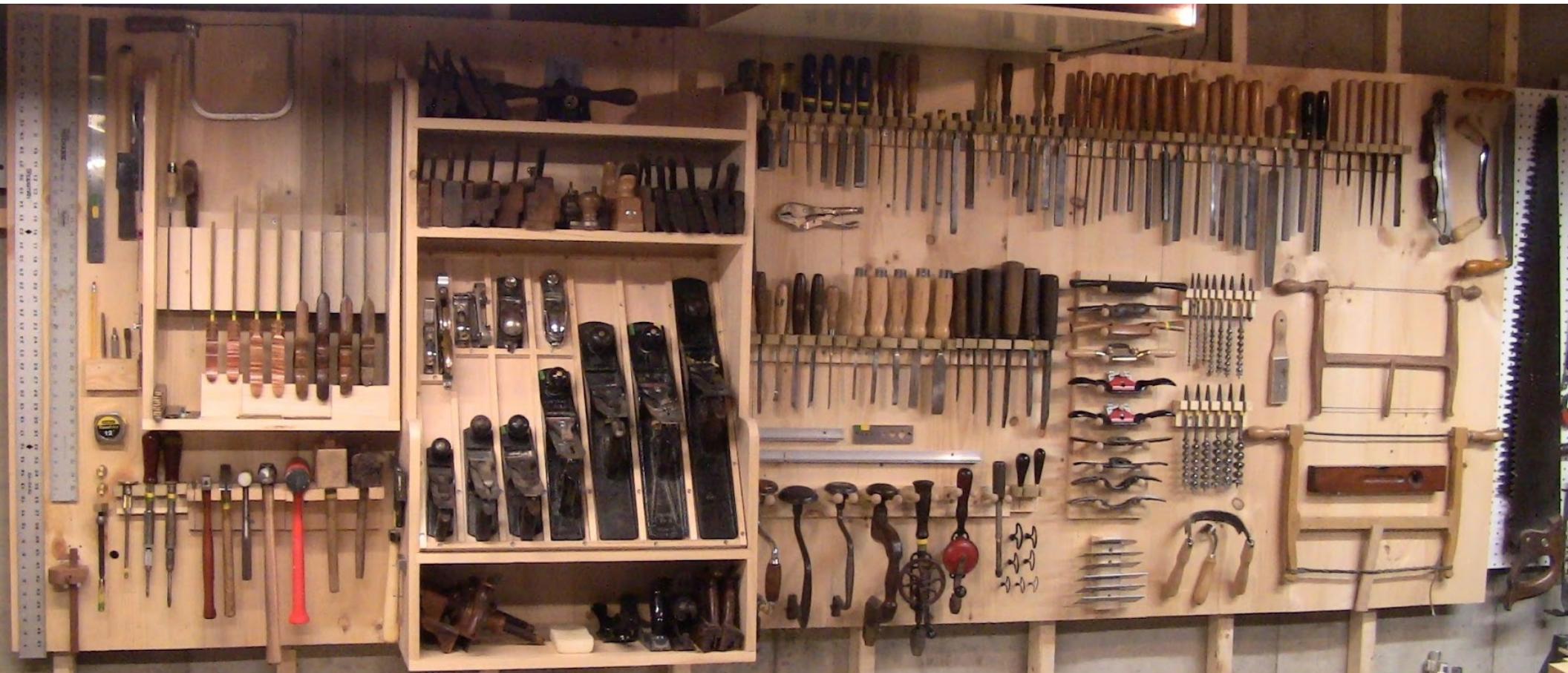
make

git and github.com

documentation: markdown



Different Tools for Different Jobs



<http://dans-woodshop.blogspot.com/>

Don't Avoid Activation Energy

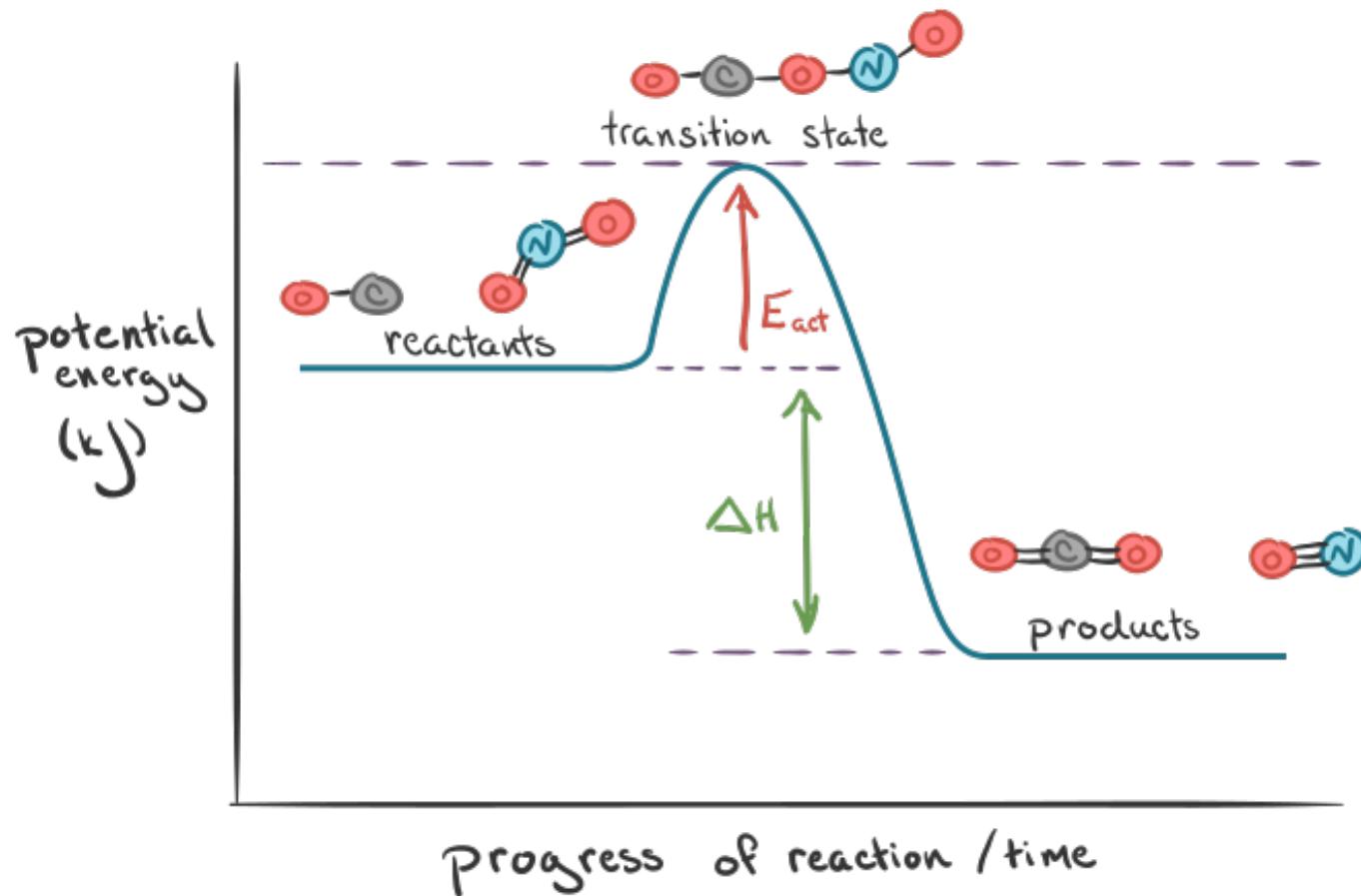


Figure from Khan Academy

<https://www.khanacademy.org/test-prep/mcat/chemical-processes/thermochemistry/a/endothermic-vs-exothermic-reactions>

Don't Avoid Activation Energy

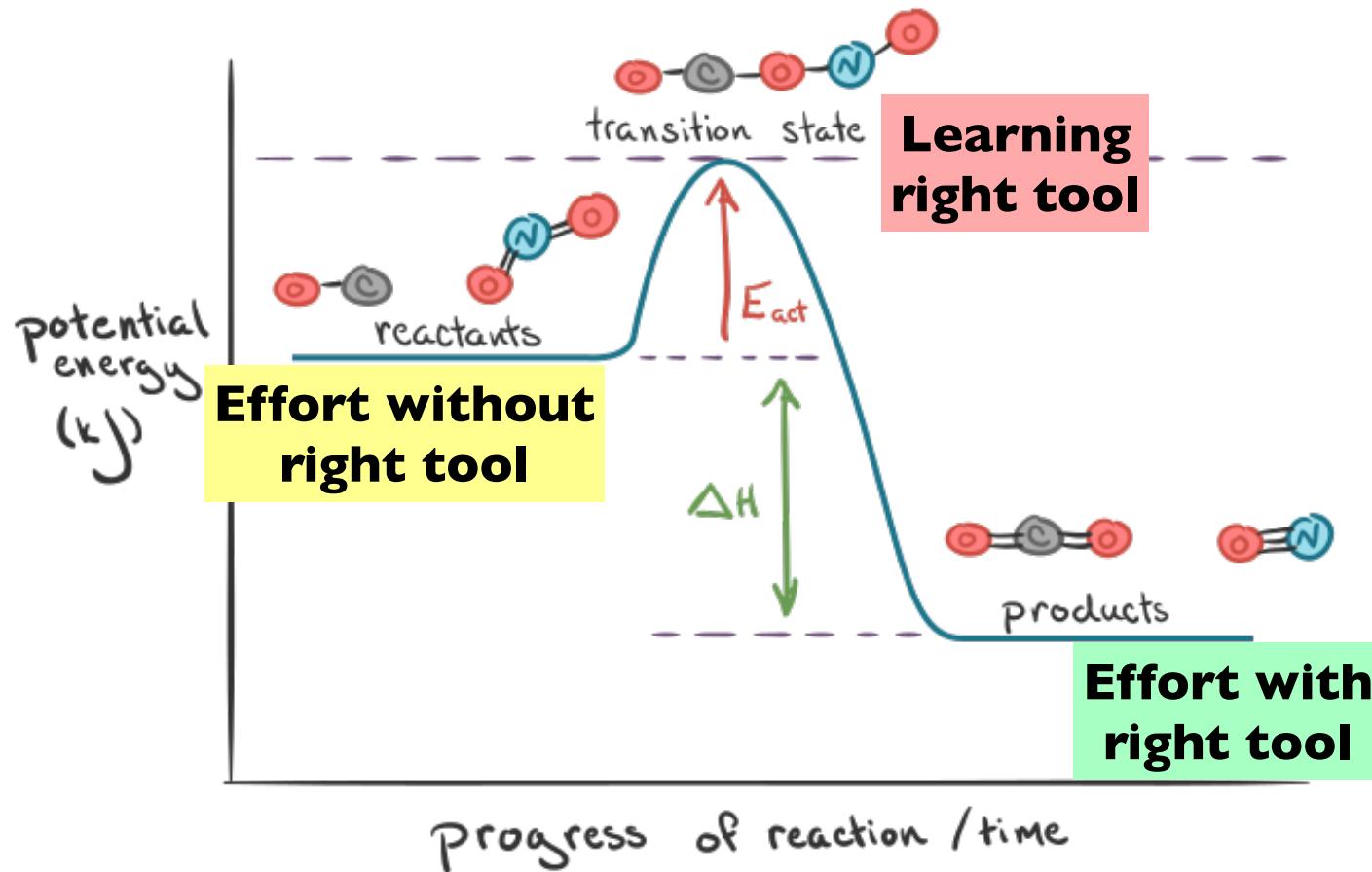
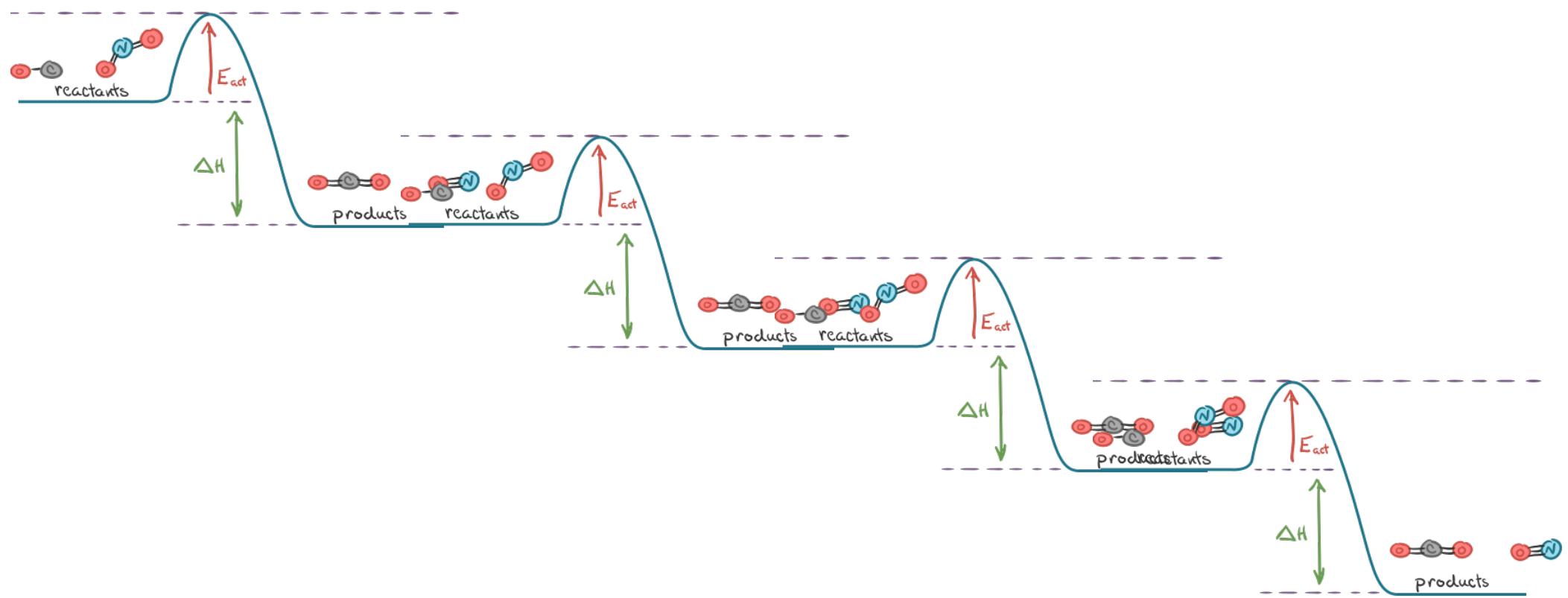


Figure from Khan Academy

<https://www.khanacademy.org/test-prep/mcat/chemical-processes/thermochemistry/a/endothermic-vs-exothermic-reactions>



It never ends... 1000x improvements possible!

Jeff Dean
Mark Horowitz over dinner
Radio SPI bug

Figure from Khan Academy

<https://www.khanacademy.org/test-prep/mcat/chemical-processes/thermochemistry/a/endothermic-vs-exothermic-reactions>

A close-up photograph showing a person's hands working on a piece of wood. The person is using a chisel to shape a rectangular block of wood, which is resting on a larger wooden board. The background shows a workshop environment with various tools and equipment.

Practice, Practice, Practice

Organized Development Environment



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

Course Schedule

§1 Bare Metal Programming

- ARM architecture and assembly language
- C functions and pointers
- Serial communication
- Linking and loading
- Memory allocation

§2 Build a Personal Computer

- Keyboard
- Graphics
- Interrupts

§3 Create Your Own Project

- Sensors
- Performance

<https://cs107e.github.io/schedule/>

Weekly Cadence

Each week has a focus topic

Pair of coordinated lectures on Wed and Mon

Lab session on Wed

Assignment released Wed after lab, due following Tue at midnight

Labs

Set of guided exercises that follow up on lecture

~2 hours

Work with partner(s)

Complete exercises and check in with staff

Leave lab ready to start assignment

Lab participation is **mandatory in person**

Philosophy: lab is hands-on, collaborative, supported, fun!

Assignments

7 weekly assignments that build on each other
This is where the learning really happens!

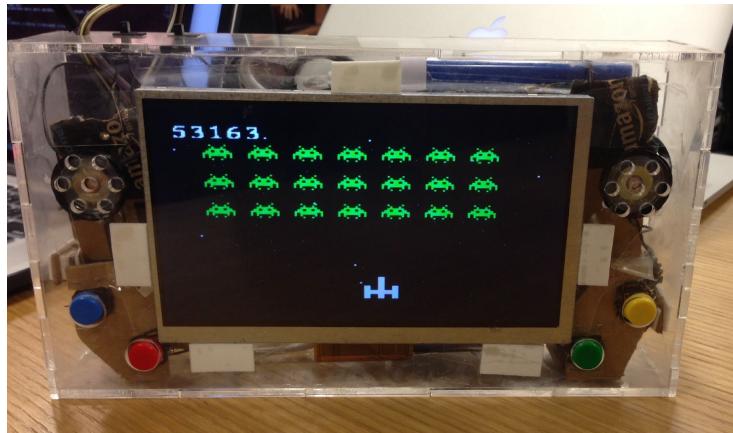
Each assignment has

- **Basic** requirement (tight spec, guided steps)
- Optional **extension** (opportunity for exploration/creativity)

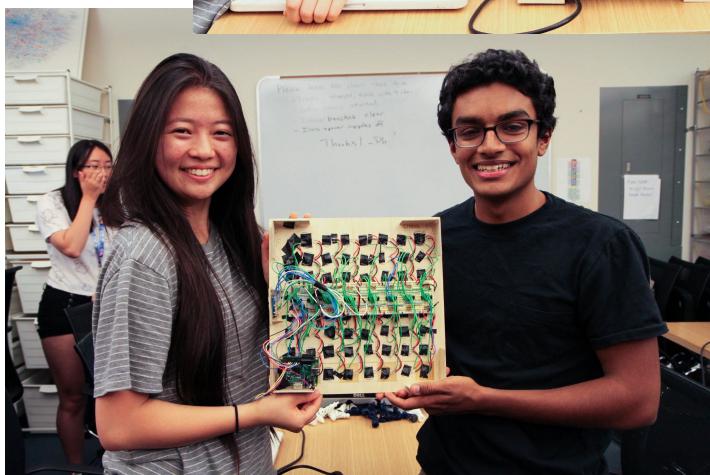
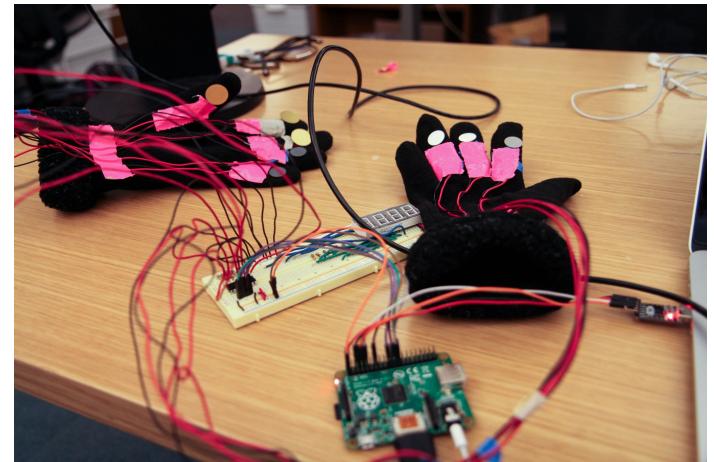
Encouraged to resubmit and correct issues in basic

End goal is complete working Pi shell of your own

Project!



project_fair.mp4



Markers for success

- Solid prerequisites: CS106B, C++, debugging
- Curiosity
- Perseverance
- Motivation

How to thrive in this course

- Consistency, follow through
- Leverage our resources, support, feedback
 - Ask questions, reach out when you need help

Interested?

FAQ cs107e.stanford.edu

Follow up discussion on Ed forum

Submit student questionnaire & contact
christos@cs.stanford.edu

We hit the ground running on March 28th!

Staff



Christos



Maria



Matt



Liana



Anna

Questions