

CS107e

Computer Systems from

the Ground Up

Christos Kozyrakis

Maria Fernandez, Matt Trost,

Liana Keesing, Anna Mistele

Spring 2022

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

<https://edstem.org/us/courses/21299/>



Christos



Maria



Matt



Liana



Anna

Learning Goal I

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

OK

```
int counter;  
int calc() {...}
```

```
int a = 20;  
unsigned int b = 6;  
if (a < b) {...}
```

Not OK

```
int calc() {  
    int counter;  
    ...  
}
```

```
int a = -20;  
unsigned int b = 6;  
if (a < b) {...}
```

OK

```
int counter;  
int calc() {...}
```

Not OK

```
int calc() {  
    int counter;  
    ...  
}
```

```
int a = 20;  
unsigned int b = 6;  
if (a < b) {...}
```

```
int a = -20;  
unsigned int b = 6;  
if (a < b) {...}
```

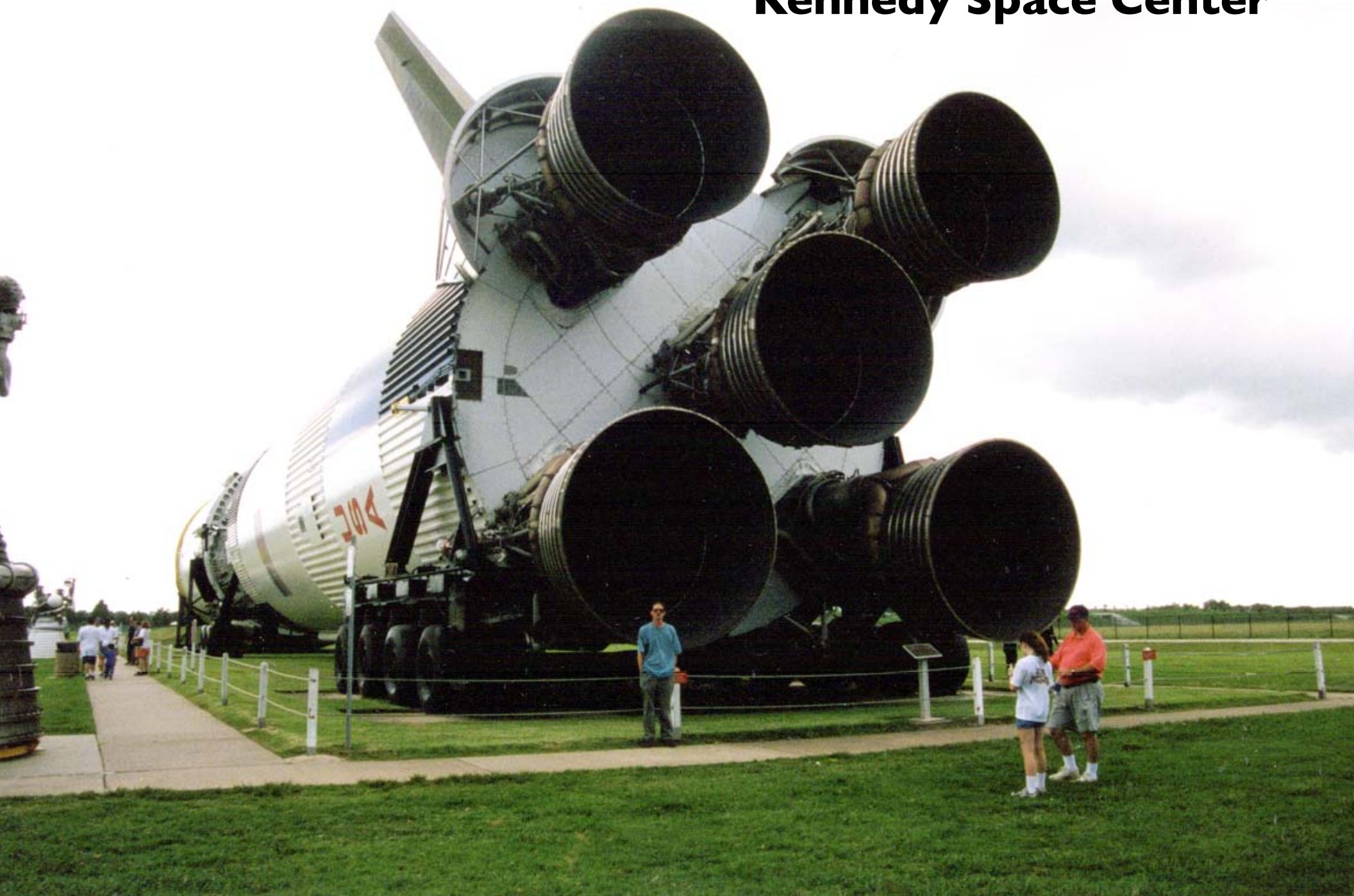
Why?

```
int main() {  
    ...  
}
```

How does your program start
at the first instruction of
main()
Or does it really?

Understanding is Empowering

Saturn V Kennedy Space Center



Falcon 9





Engineer for Excellence!

Perseverance!

First steps are often the hardest

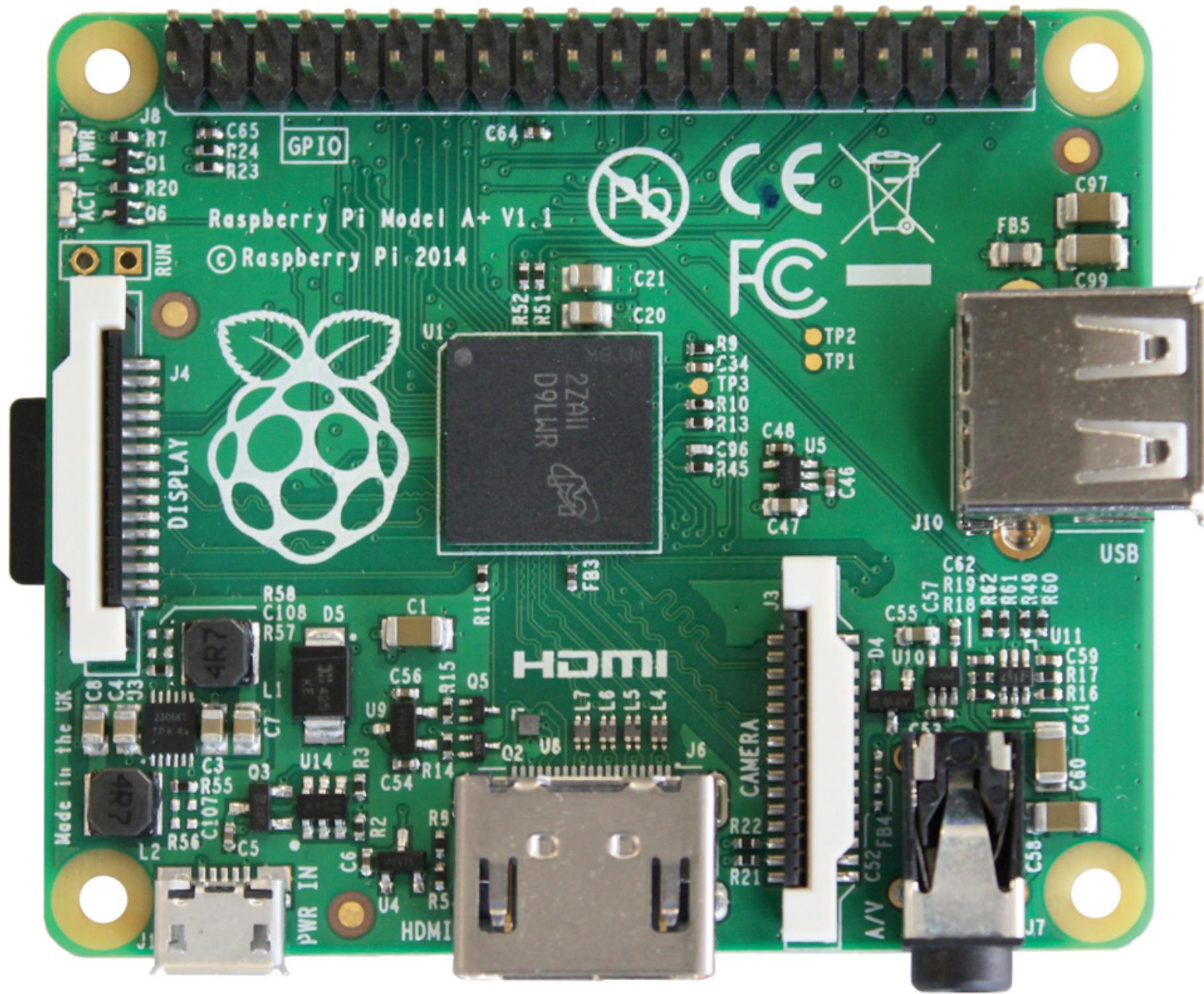
That's why we're here!

Bare Metal on the Raspberry Pi

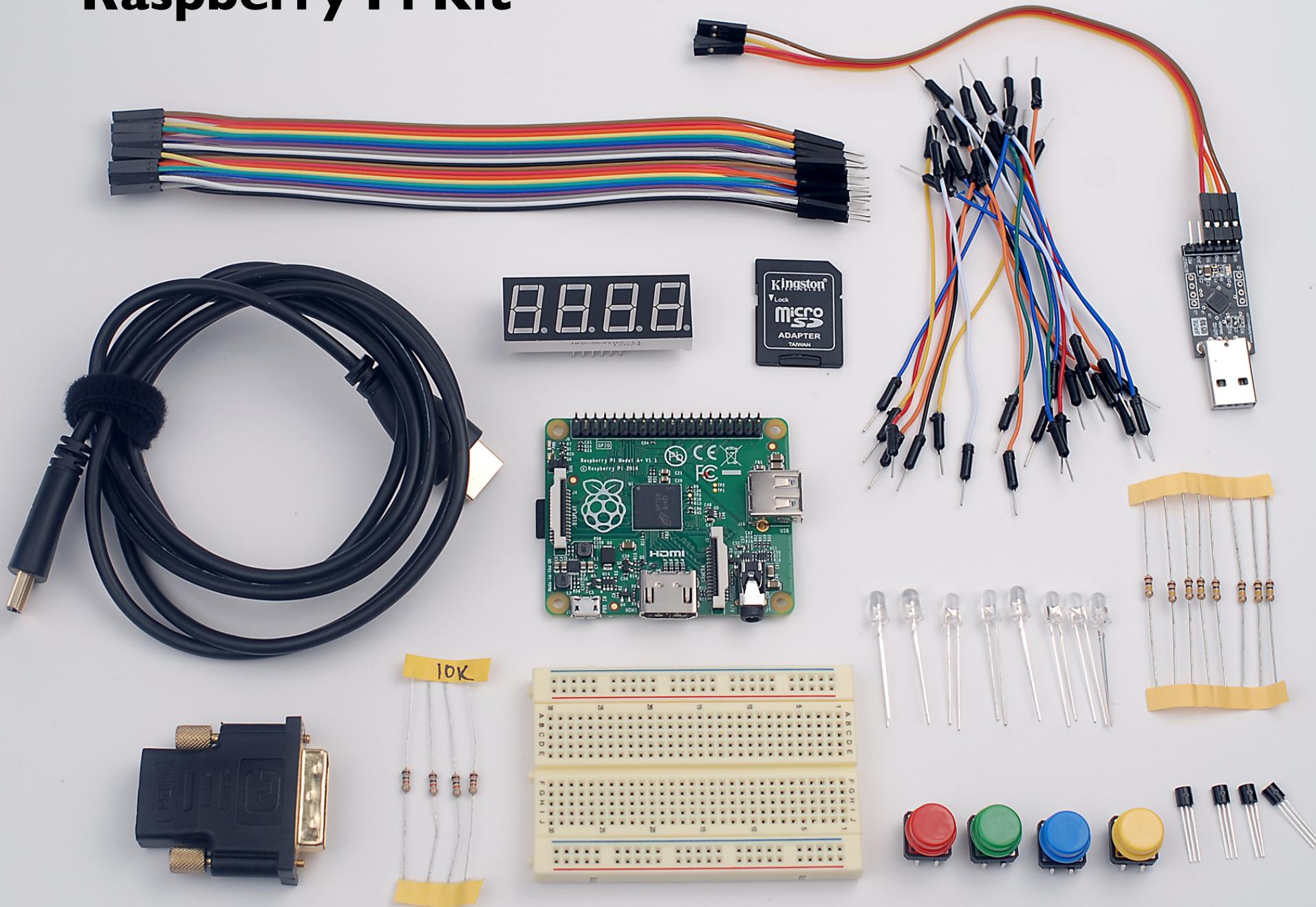
Definition: Bare metal programming involves no operating system (programmer constructs libraries)

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

You'll understand every line of code in the system.

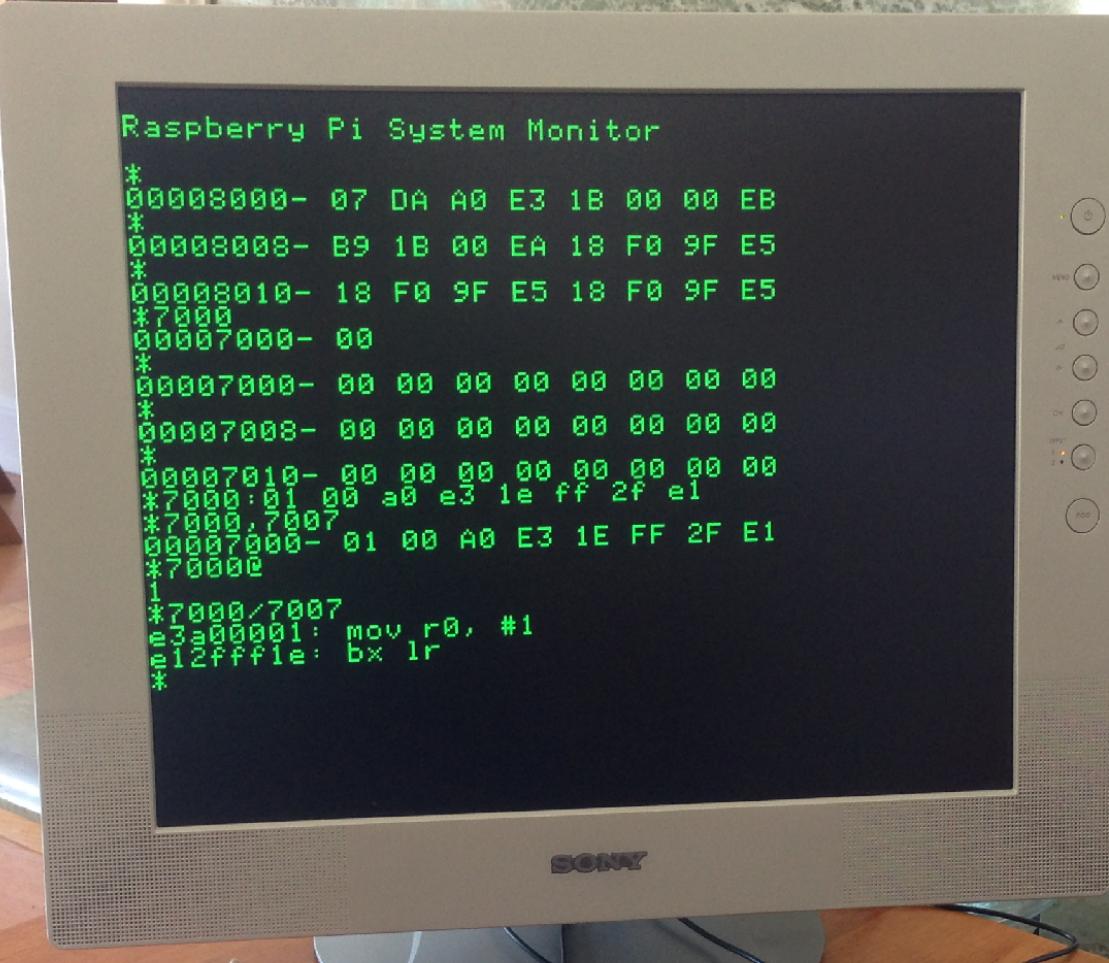
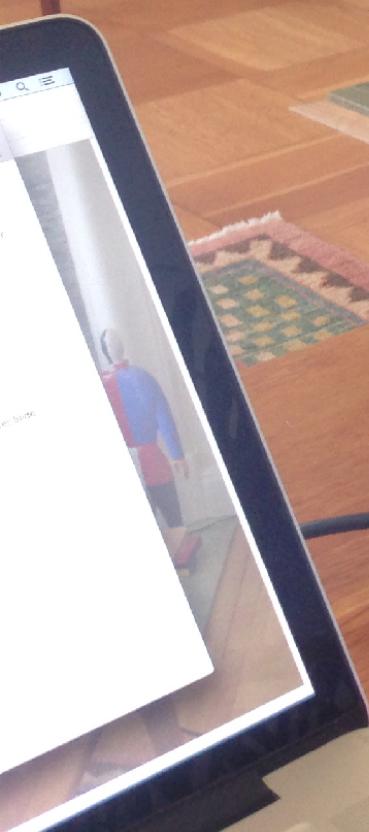


Raspberry Pi Kit



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  
*
```



**Almost every instruction
will be code you've written!**

Learning Goal 2

**Master your tools
Learn their value**

Software Tools

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

Programming languages: C, ...

gcc

as

ld

binutils: nm, objcopy, objdump, ...

make

git and github.com

documentation: markdown



Different Tools for Different Jobs



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

Organized Development Environment



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

Don't Avoid Activation Energy

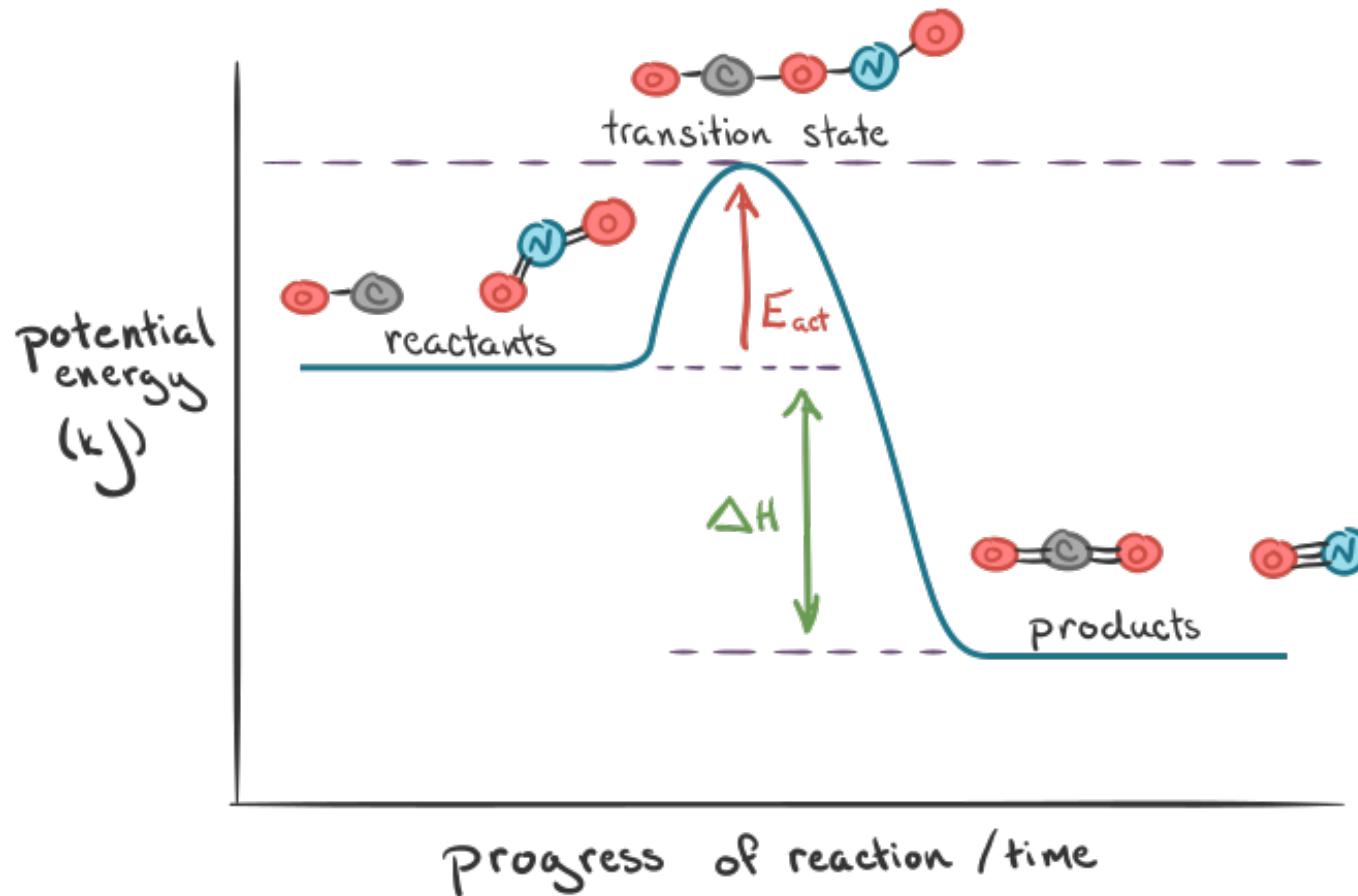


Figure from Khan Academy

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

Don't Avoid Activation Energy

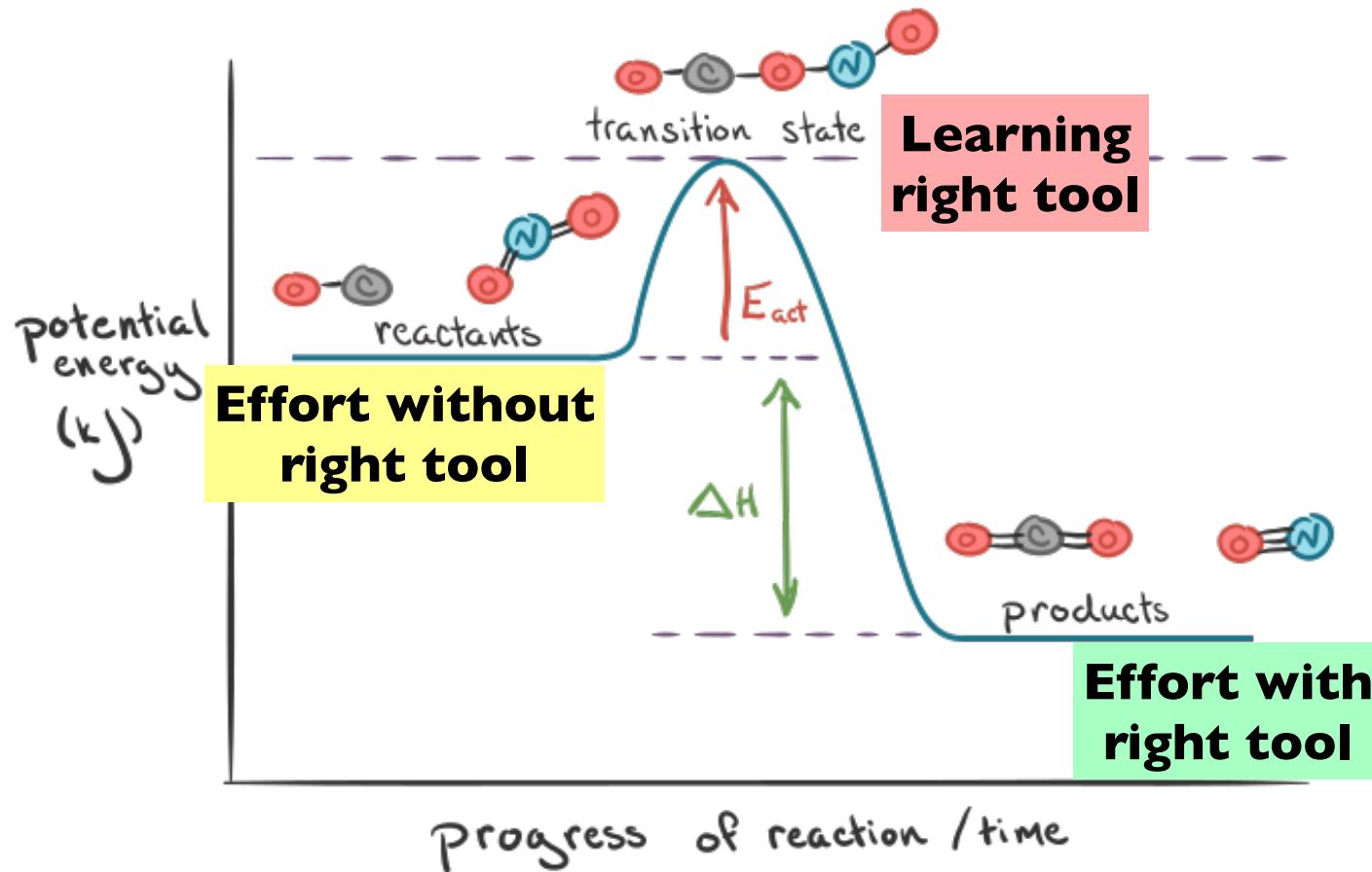


Figure from Khan Academy

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

Hyperbolic Discounting



<https://medium.com/behavior-design/hyperbolic-discounting-aefb7acec46e>

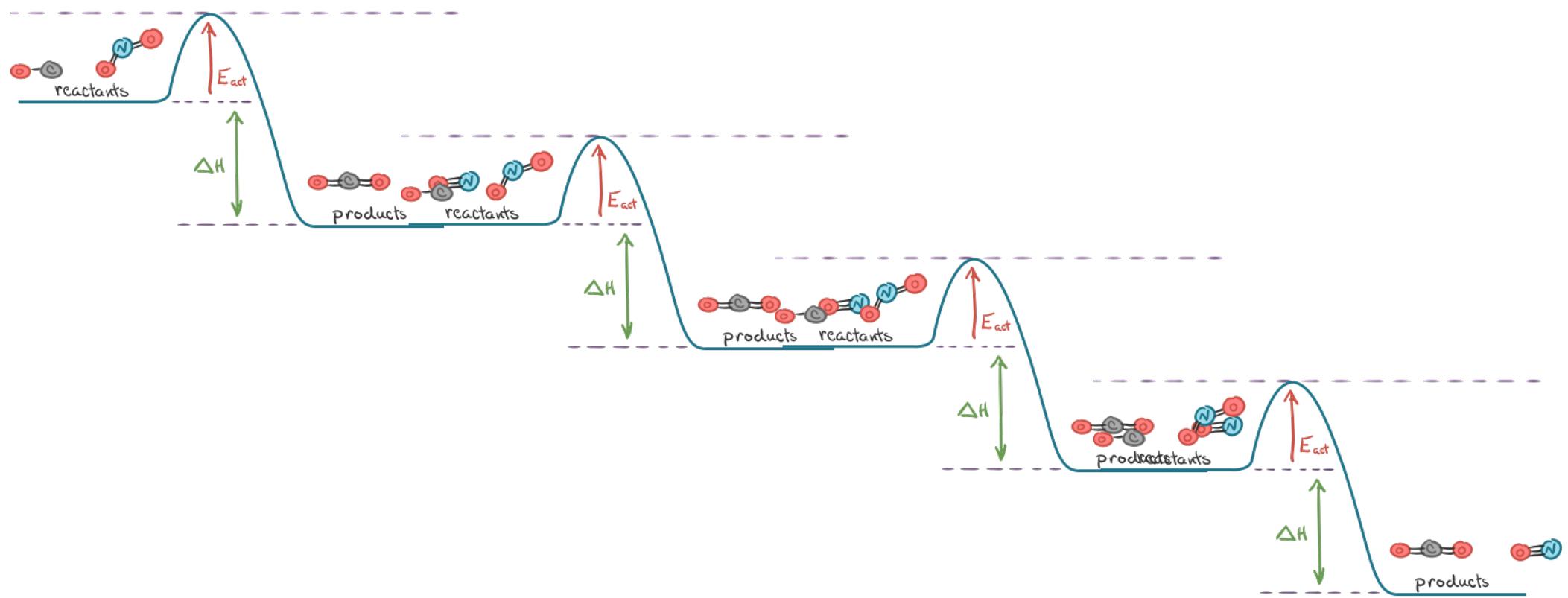
Hyperbolic Discounting



<https://medium.com/behavior-design/hyperbolic-discounting-aefb7acec46e>

A close-up photograph showing a person's hands working on a piece of wood. The person is using a chisel to create a rectangular cutout in a dark, rectangular block of wood. The wood is held in place by several metal clamps. The person's hands are steady, focused on the task. In the background, there's a bottle of water and some workshop equipment.

Learn & Practice, Practice, Practice



It never ends... 1000x improvements possible!

Figure from Khan Academy

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

Debugging and Troubleshooting



Course Topics

cs107e.github.io

§ I Bare Metal Programming

1. ARM processor and memory architecture
2. ARM assembly language and machine code
3. C
4. Functions
5. Serial communication
6. Linking and loading
7. Memory allocation

§2 Personal Computer

- I. Keyboard**
- 2. Graphics**
- 3. Interrupts**

Goal: Build Raspberry Pi shell

§3 Additional Topics

- I. Sensors**
- 2. Performance**

And a couple of special/guest lectures

Administration

Weekly Cadence

Each week has a focus **topic**

Pair of coordinated **lectures** on Wed and Mon

Lab on Tue

Assignment handed out Tue after lab, YEAH session Wed, due following Tuesday 5pm

Staying on pace leads to best outcomes!

Lectures

Attendance is **necessary**

Content is unique to our course, no textbook
The readings/slides are not a standalone resource

Attendance allows you to participate, ask questions, stay on schedule

Pose questions verbally or type in chat, but **do ask!!**

Labs

Attendance is **mandatory**

Guided exercises, work with peers, **check in** with staff

Finish lab **ready** for assignment, esp. experience with tricky parts (hardware/software interface)

Philosophy: lots-of-help, hands-on, collaborative

Lab room: Gates B02

Assignments

7 weekly assignments

Build on each other, complete full system

Assignment specifications

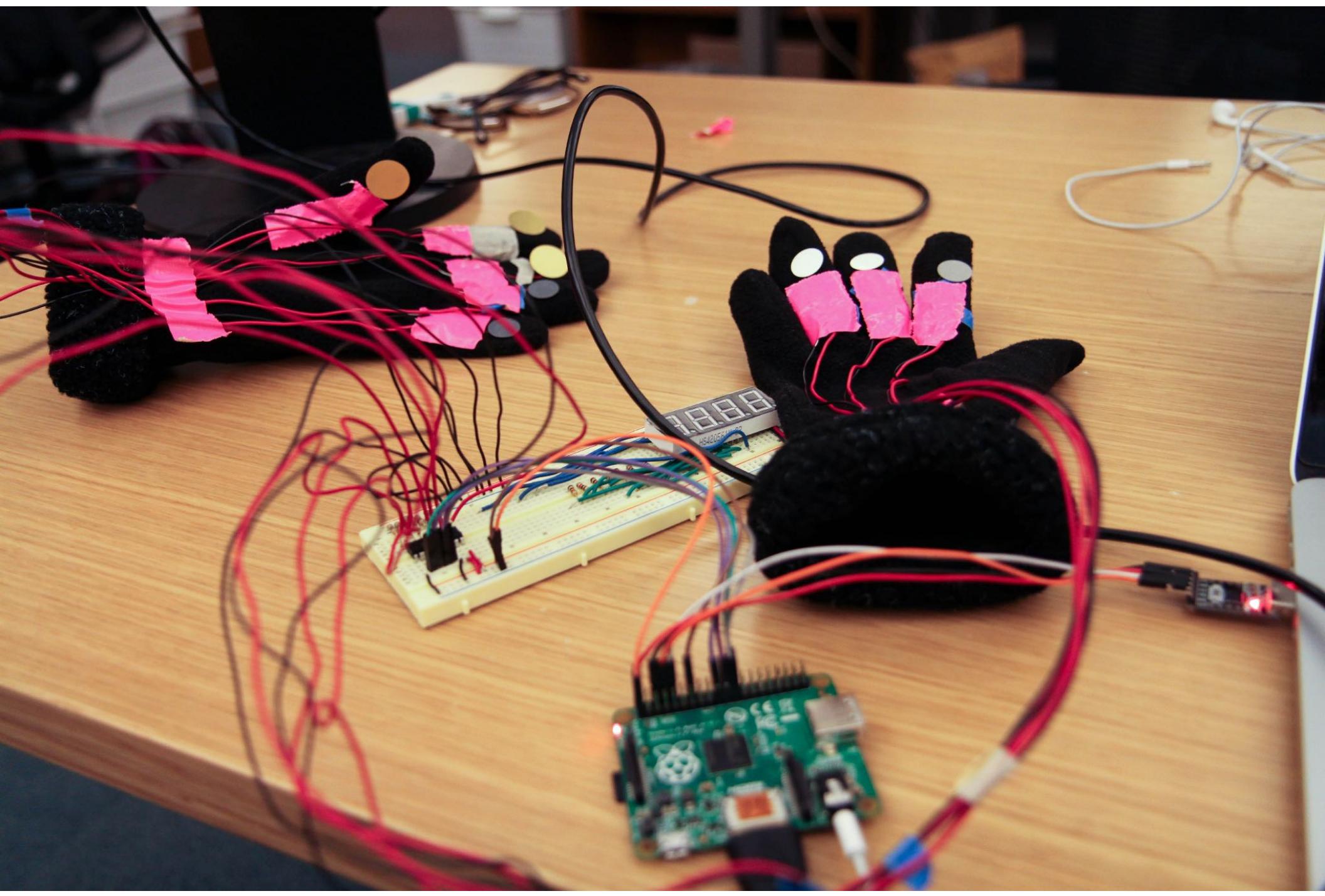
Core (required, tight spec, guided steps)

Extension (optional, opportunity for your exploration/
creativity)

Revise and **resubmit** to address issues in core functionality

Project

Design and build **your own system**

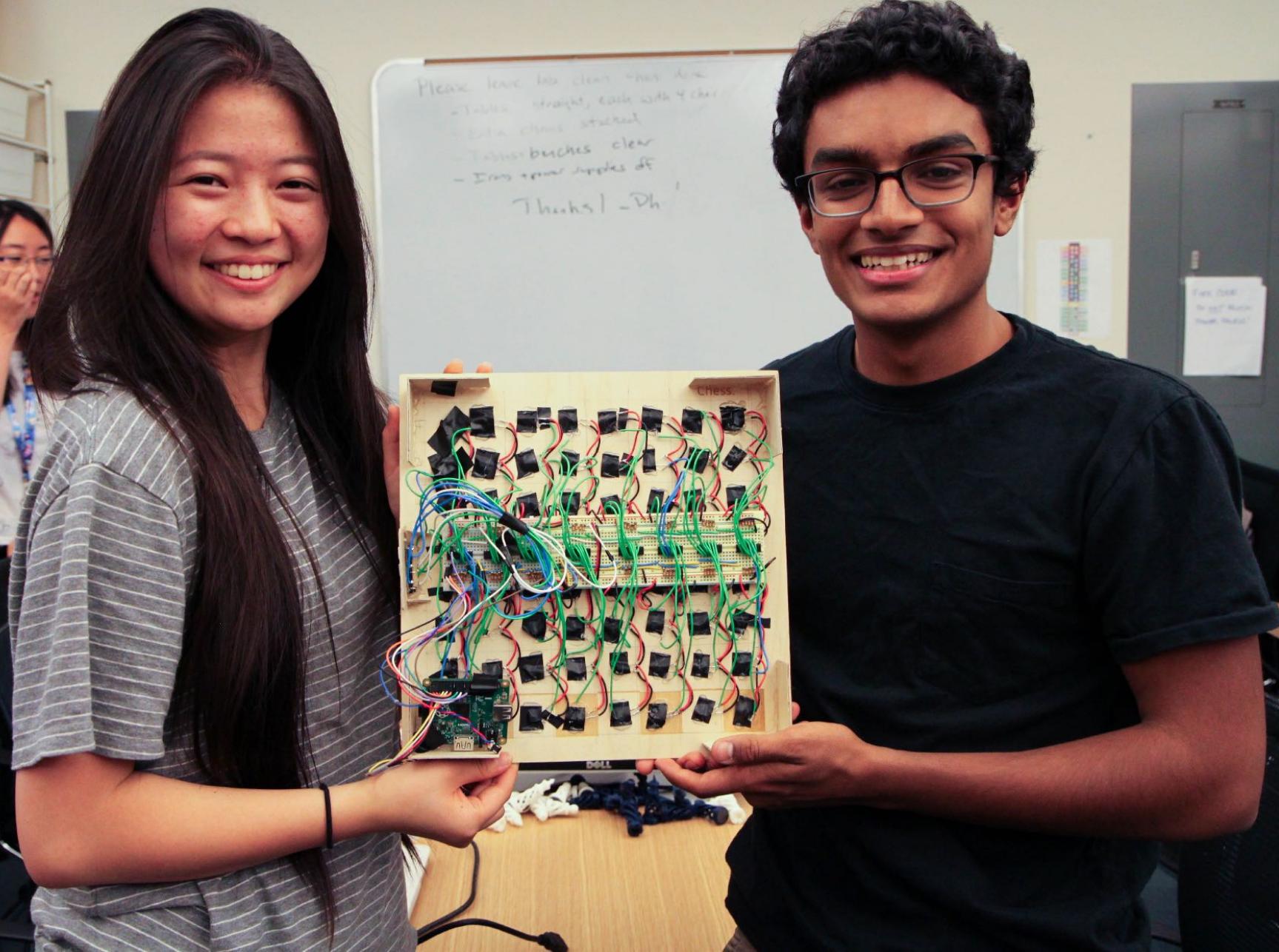


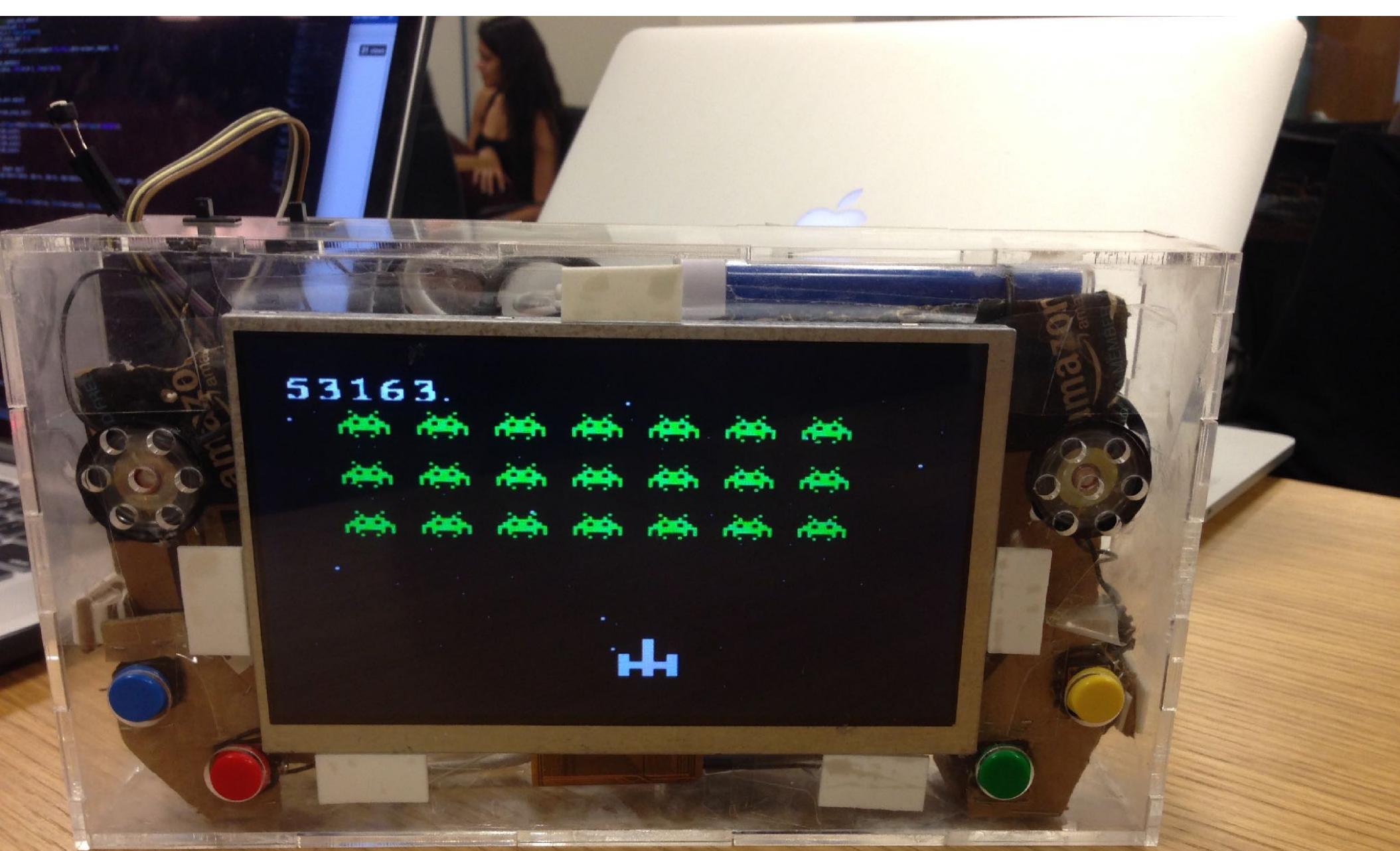




Please leave box clean - no dust
- Tables straight, each with 4 chairs
- Extra chairs stacked
- Tissue boxes clear
- Irons & paper supplies off

Thanks! - Dh





Learning community

Stay **connected**

Participate in lecture

Collaborate in lab

Discuss on Ed forum

Come to office hours

Meet up in lab room

Be **curious**. Learn by **doing**. Ask for and offer **help**.

Dealing with COVID

The Washington Post
Democracy Dies in Darkness

Philip Levis

Health

Parenting a child under 12 in the age of delta: 'It's like a fire alarm every day'

Routine outings become tricky decisions for the youngest Americans, who are still ineligible for coronavirus vaccines

Screenshot

I have small kids (not vaccinated yet)

Delighted to be teaching in person but need to be careful

Stanford requires masks are worn in lectures
- Please wear them well and carefully! Even if you are alone in lab.

First Week

Today

Fill out course application if you haven't already

Make sure you are registered on Axess

Fill in form for github IDs

Assignment 0

Join forum [https://edstem.org/us/courses/21299/
discussion/](https://edstem.org/us/courses/21299/discussion/)

Read and understand our guides on basic topics:
electricity, numbers, and UNIX

Create github account and send us your GitHub id

Install/setup your development environment

Number Representations

Binary representation

Hexadecimal

Bit operators

Guide: <https://cs107e.github.io/guides/numbers/>

Basic Electricity

Voltage and current

Ohms Law : $V = I R$

Power : $P = IV$

Driving an LED

Transistor switches

Breadboarding

Guide: <https://cs107e.github.io/guides/electricity/>

Unix Command Line

Moving around the file system

Creating, moving, and deleting files

Compiling and running programs

Profiles and paths

Guide: <https://cs107e.github.io/guides/unix/>

Watch cs107 UNIX videos!

How to Succeed in CS107e

Matt
CS107e Spring 2022

Keep Pace with the Class

- Weekly cadence.
 - Tough to catch up if you fall behind.
- Labs are Tuesdays 7-9pm.
- Assignments released after lab.
- Assignments due Tuesdays 6pm (before lab).
 - 48-hour “grace period” until Thursday 6pm.
- **START EARLY**
 - Take advantage of YEAH hours, weekday OH (no weekend OH)

Understand the Assignment Before Starting to Code

- Half the battle is reading and understanding the assignment spec.
- YEAH Hours are a great way to start!
- Write out logic on paper/whiteboard first.
 - e.g. backtrace
- Reference the header files →

```
/*
 * 'gpio_set_function'
 *
 * Set a GPIO function for GPIO pin number 'pin'. Settings for other pins
 * should be unchanged.
 *
 * @param pin      the GPIO pin number to configure
 * @param function the GPIO function to set for the pin
 *
 * If 'pin' or 'function' is invalid, does nothing.
 */
void gpio_set_function(unsigned int pin, unsigned int function);
```

Collaborate in Allowed Ways

- Form study groups! Help each other understand course concepts, C syntax, and assignment specs.
 - Gates B02 (our lab room) usually becomes a community space!
- If you don't understand something, ask and clarify it before continuing!
 - Coding without understanding is a great way to produce very buggy code...
- The course staff is here to help!
 - Leverage office hours and ask questions in lecture
 - Ask conceptual questions on Ed
 - If you start to fall behind, or have exceptional circumstances, reach out to the course staff! We want to work with you.

Clarifications about Collaboration

- From the course policies: <https://cs107e.github.io/policies/>
- You must **cite** when you:
 - Discuss assignment design (your approach, not the spec.)
 - Get suggestions from another student for debugging.
 - Share or brainstorm testing strategies or inputs.
 - Use code from lab or other sources.
- You may **not**:
 - Look at other students' code.
 - Jointly develop an assignment design or jointly write code.
 - Jointly debug an assignment.
 - Use external resources for assignment-specific code.

How to Ask for Help

- Ed is for logistical and conceptual questions.
- All questions specific to your code should be addressed in office hours.
- In office hours, be able to answer:
 - What is the expected behavior of the problematic section of code?
 - What is the observed behavior?
 - Are there any parts of the spec, or C syntax, that you are unsure about?
- Many questions we answer in OH are resolved without us needing to look at the code, but instead clarifying the spec or a lecture concept.

Keep Up-to-Date on Ed

- We use Ed to post:
 - Important logistical announcements.
 - Specific assignment clarifications or tips.
 - Guides and walkthroughs of difficult concepts.
 - Excellent references for tooling or concepts.
- Don't miss it!

Coding Tips

- First, understand the high level logic of what you need to do.
 - Sketching out on paper might be helpful.
- Translate the high level logic into discrete steps, then comment those steps in your function!
 - Makes your code easier to read, debug, and audit.
 - Logic problem or implementation problem?
- Then for each block comment, code the implementation.
 - Code in small chunks, and test after each chunk if possible.

Block Commenting: an Example

```
72
93  ↳ void RaftLog::RemoveEntriesAtAndAfter(uint64_t index)
94  {
95      current_logslice_filepath = LOG_BASE_PATH;
96      uint32_t num_log_slices = index / MAX_ENTRIES_PER_SLICE;
97      current_logslice_filepath += std::to_string( val: num_log_slices);
98      logslice_storage =
99          PersistentProtoStorage<LogSlice>( destination: current_logslice_filepath);
100     logslice_storage.Restore( &: current_logslice);
101     uint64_t slice_offset = index % MAX_ENTRIES_PER_SLICE;
102     auto erase_start_iter :RepeatedPtrIterator<...> = current_logslice.mutable_entries()->begin() + slice_offset;
103     current_logslice.mutable_entries()->erase(
104         first: erase_start_iter,
105         last: current_logslice.mutable_entries()->end());
106     logslice_storage.Save( data: current_logslice);
107     log_info.set_num_log_slices( value: num_log_slices);
108     loginfo_storage.Save( data: log_info);
109 }
```

```
93  ↵  void RaftLog::RemoveEntriesAtAndAfter(uint64_t index)
94  {
95      current_logslice_filepath = LOG_BASE_PATH;
96      uint32_t num_log_slices = index / MAX_ENTRIES_PER_SLICE;
97      current_logslice_filepath += std::to_string( val: num_log_slices);
98      logslice_storage =
99          PersistentProtoStorage<LogSlice>( destination: current_logslice_filepath);
100
101     logslice_storage.Restore( &: current_logslice);
102     uint64_t slice_offset = index % MAX_ENTRIES_PER_SLICE;
103     auto erase_start_iter : RepeatedPtrIterator<...> = current_logslice.mutable_entries()->begin() + slice_offset;
104     current_logslice.mutable_entries()->erase(
105         first: erase_start_iter,
106         last: current_logslice.mutable_entries()->end());
107     logslice_storage.Save( data: current_logslice);
108
109     log_info.set_num_log_slices( value: num_log_slices);
110     loginfo_storage.Save( data: log_info);
111 }
```

```
93 void RaftLog::RemoveEntriesAtAndAfter(uint64_t index)
94 {
95     // Read the logslice file containing this index.
96     current_logslice_filepath = LOG_BASE_PATH;
97     uint32_t num_log_slices = index / MAX_ENTRIES_PER_SLICE;
98     current_logslice_filepath += std::to_string( val: num_log_slices);
99     logslice_storage =
100         PersistentProtoStorage<LogSlice>( destination: current_logslice_filepath);
101    logslice_storage.Restore( &: current_logslice);
102
103    // Erase all entries in this logslice at and after the index.
104    uint64_t slice_offset = index % MAX_ENTRIES_PER_SLICE;
105    auto erase_start_iter : RepeatedPtrIterator<...> = current_logslice.mutable_entries()->begin() + slice_offset;
106    current_logslice.mutable_entries()->erase(
107        first: erase_start_iter,
108        last: current_logslice.mutable_entries()->end());
109
110    // Save the logslice and update the number of loglices.
111    logslice_storage.Save( data: current_logslice);
112    log_info.set_num_log_slices( value: num_log_slices);
113    loginfo_storage.Save( data: log_info);
114 }
```

Debugging Tips

- Hardware problem or software problem?
 - Are all your wires connected properly? Are any too loose?
 - Does your code work on your friend's Pi/breadboard?
- Find the simplest/smallest test case that triggers the error.
- Walk through the code, either manually or with gdb.
- If you're unsure what the behavior should be, you can check what the staff implementation does.
 - Ask staff to show you how.

Invest in Learning Your Tools

- Vim, Emacs, Git, and the Unix command line have high learning curves.
 - I will host a couple special OH sessions to introduce these tools.
 - I will also post some great reference resources on Ed.
- If you're doing something manually and tediously, there's almost certainly a fast way to do it.
 - Ask staff for advice!
- Don't feel that you need to be a tooling master

Do NOT...

- Modify the header files.
 - Our grading framework uses the original header files, so it won't reflect your modifications.
- Deviate from the spec.
 - Our grading framework expects spec behavior.
- Skip lab.
 - Lab attendance is graded.
 - Lab is critical to understanding the assignment.
- Let your Pi rest on metal! All the pins are exposed...

Questions?