

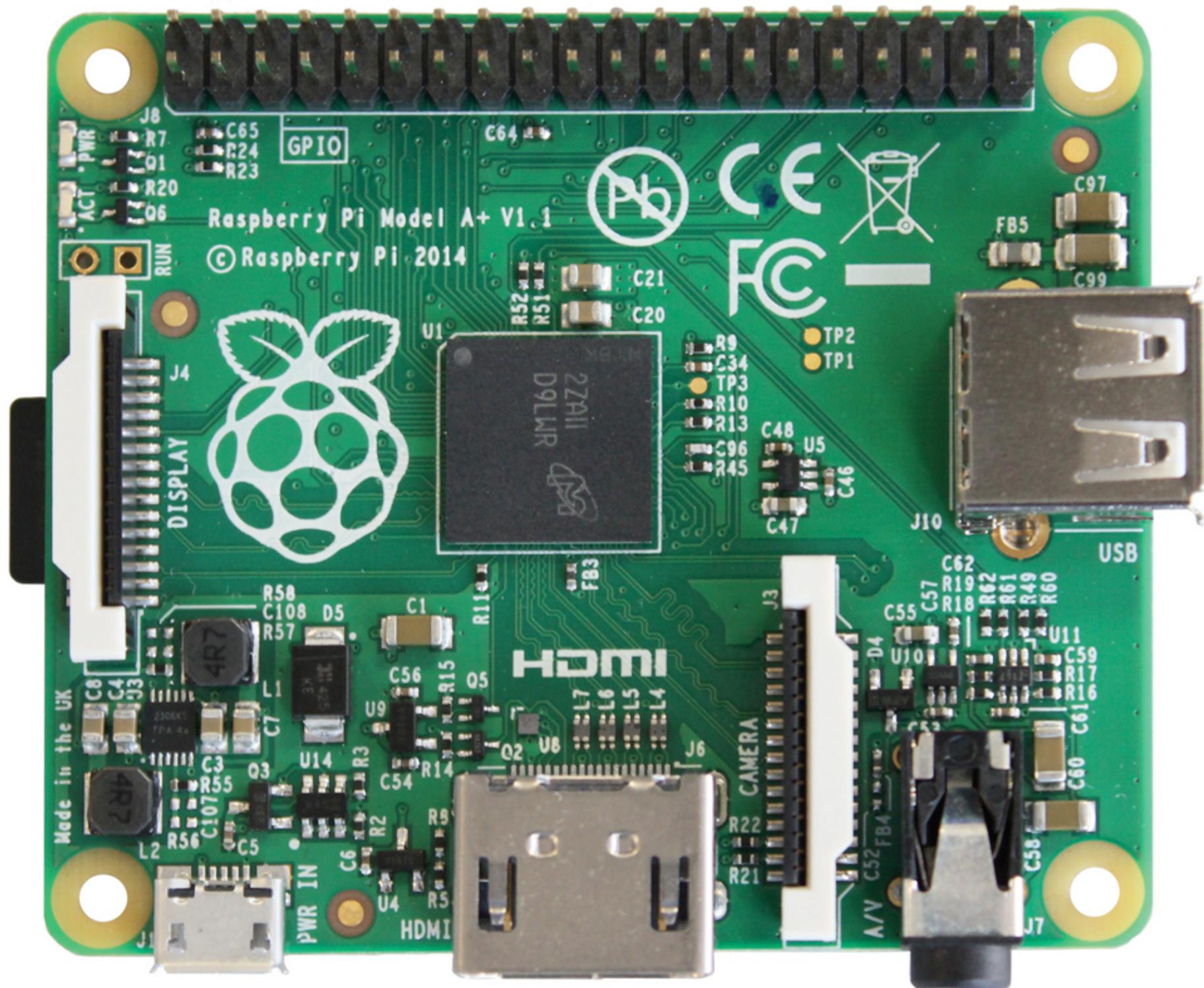
# **Computer Systems from the Ground Up**

**Pat Hanrahan, Dawson Engler, Phil Levis**

**Isabel Bush, Jane E,  
Omar Rizwan, Anna Zeng**

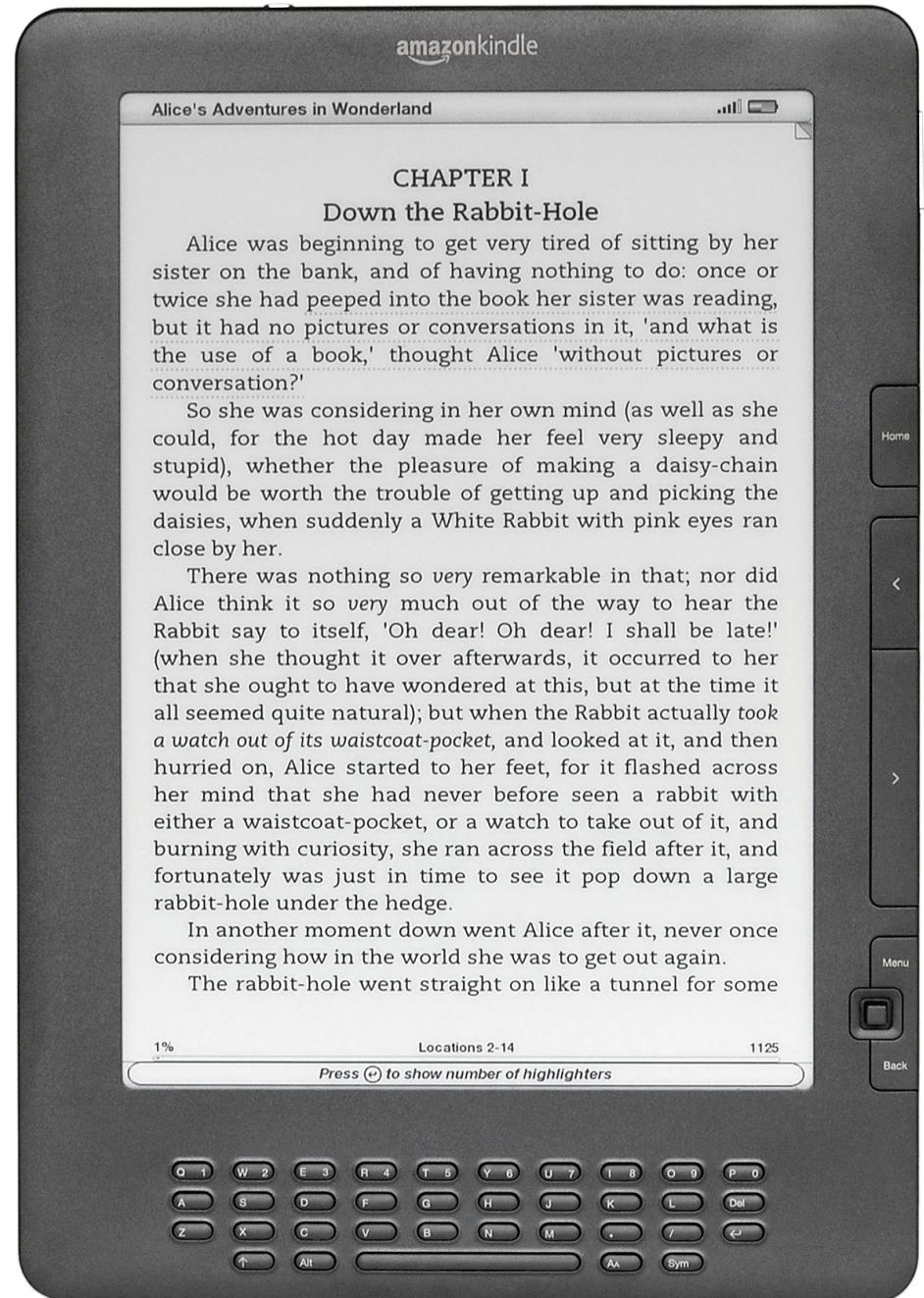
# **Learning Goal 1**

**Understand how computers  
represent data and execute programs**





**iPhone 3**



**Kindle 2**



# **Raspberry Pi System Monitor**



# **Bare Metal on the Raspberry Pi**

**Definition:** Bare metal programming involves no operating system and minimal use of libraries.

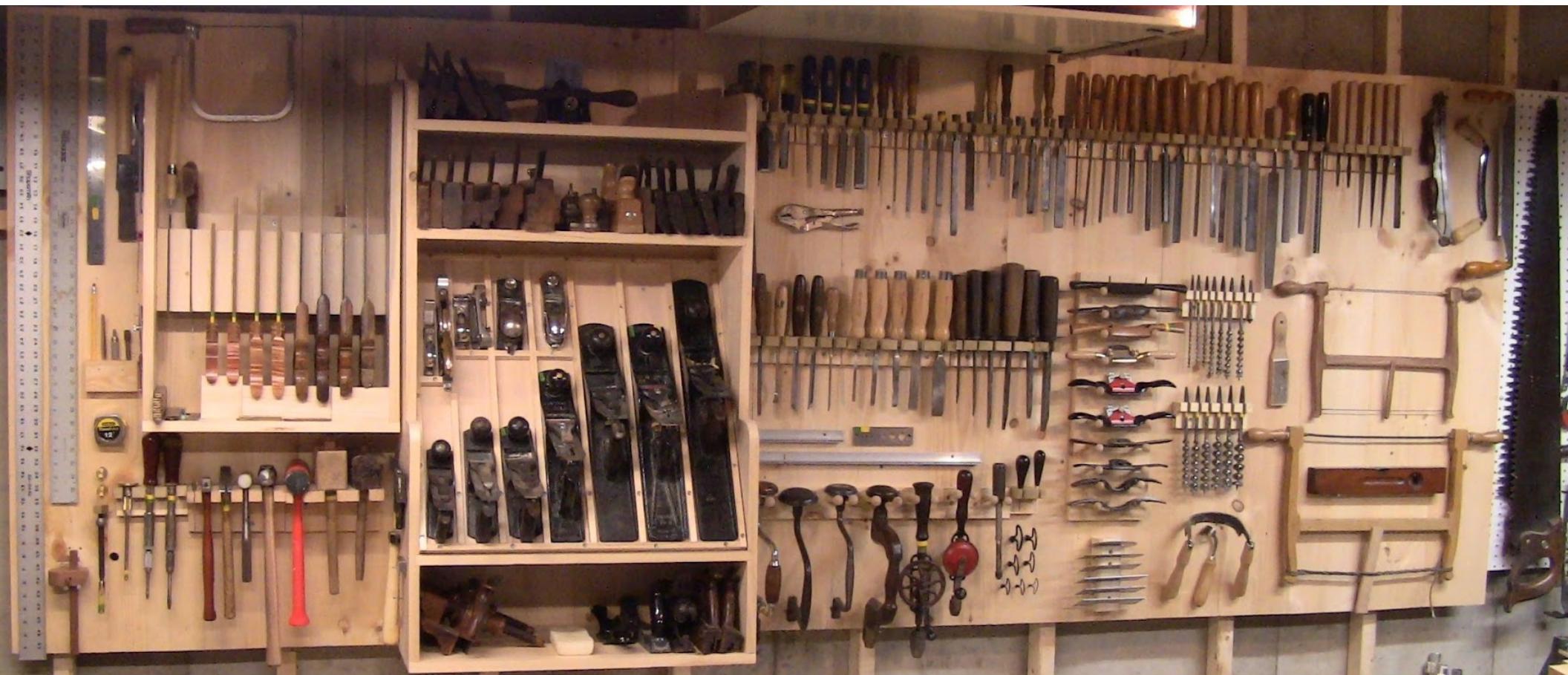
**Bare metal programs boot and startup on their own, and directly control peripherals.**



# **Learning Goal 2**

## **Master your tools**

# Different Tools for Different Jobs



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

# Organize Your (Dev) Environment



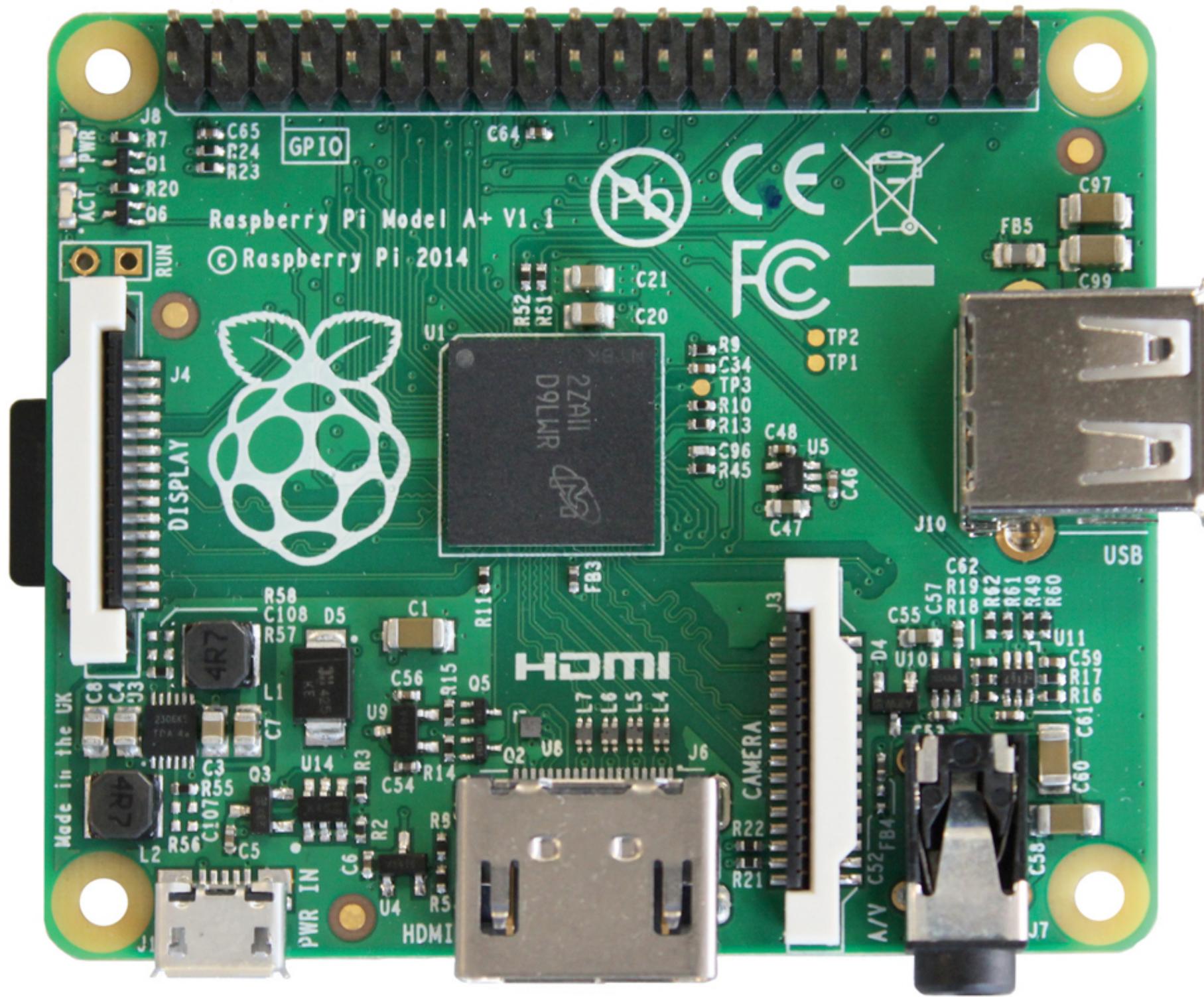
<http://amhistory.si.edu/juliachild/>



**Mastery**

# **Exercise**

## **Inspect Raspberry Pi**



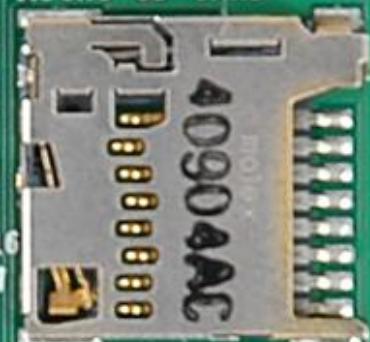
AKCE MC1  
V-OF3  
1439 1-6

PP35  
PP22  
PP23  
PP27

R12 C40 C17  
C36 C95 F82 C69  
C94 C51 C49 C18 C37  
C50 C9 F51 C14 R25  
C45 C29 C13 C12 C35  
C30

TRST\_N  
TDI  
TDO  
TMS  
TCK  
GND  
PP32 PP29 PP34  
PP33 PP31

**MICRO SD CARD**



PP10  
PP13

27/10

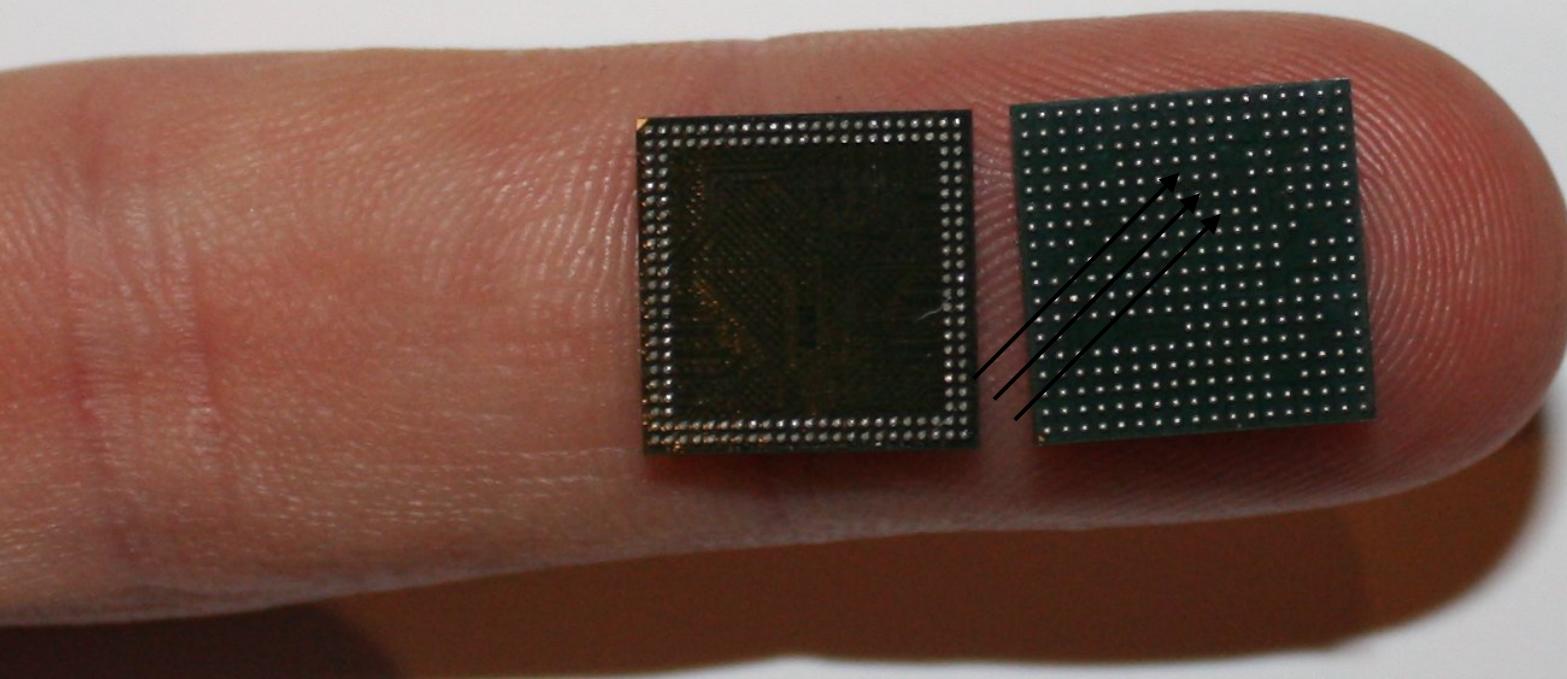
1

J9

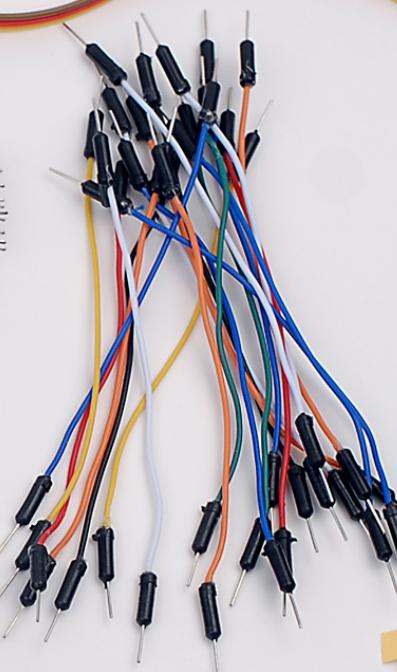
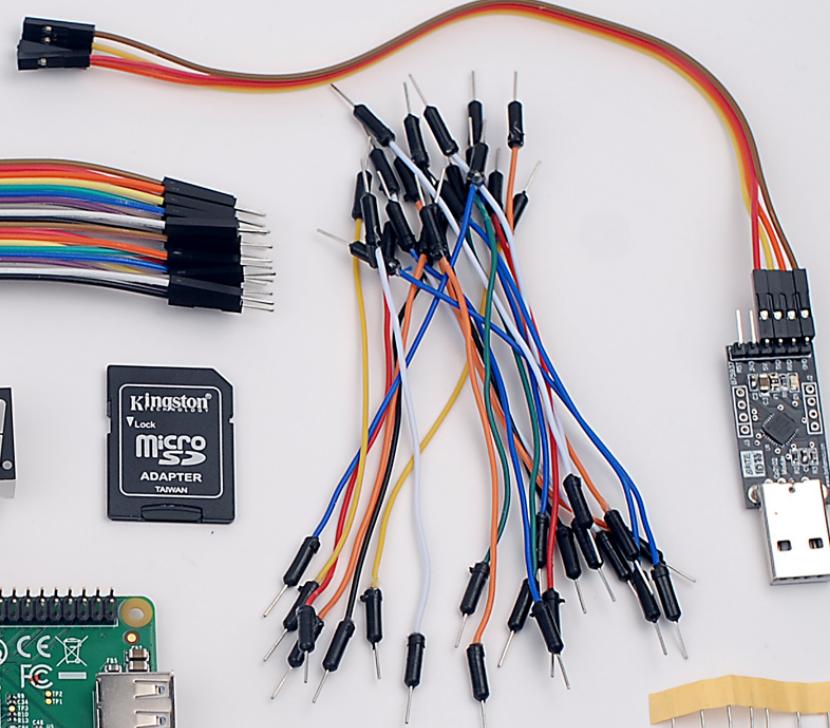
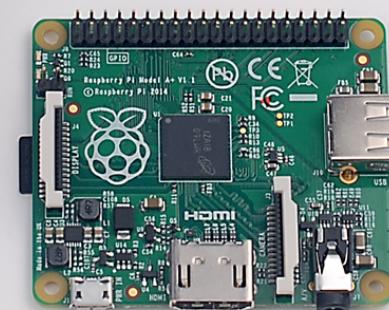
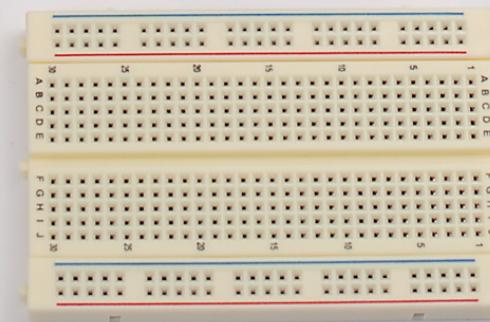
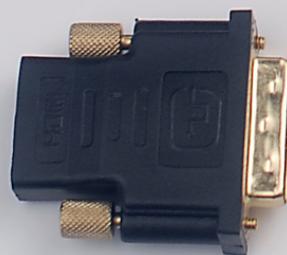
A close-up photograph of a green printed circuit board (PCB) showing several surface-mount components. The components are labeled with codes such as C2, R1, PP8, PP4, PP9, PP7, PP1, PP2, and PP3. The labels are placed near their respective components, which appear as small orange or grey rectangles with leads.

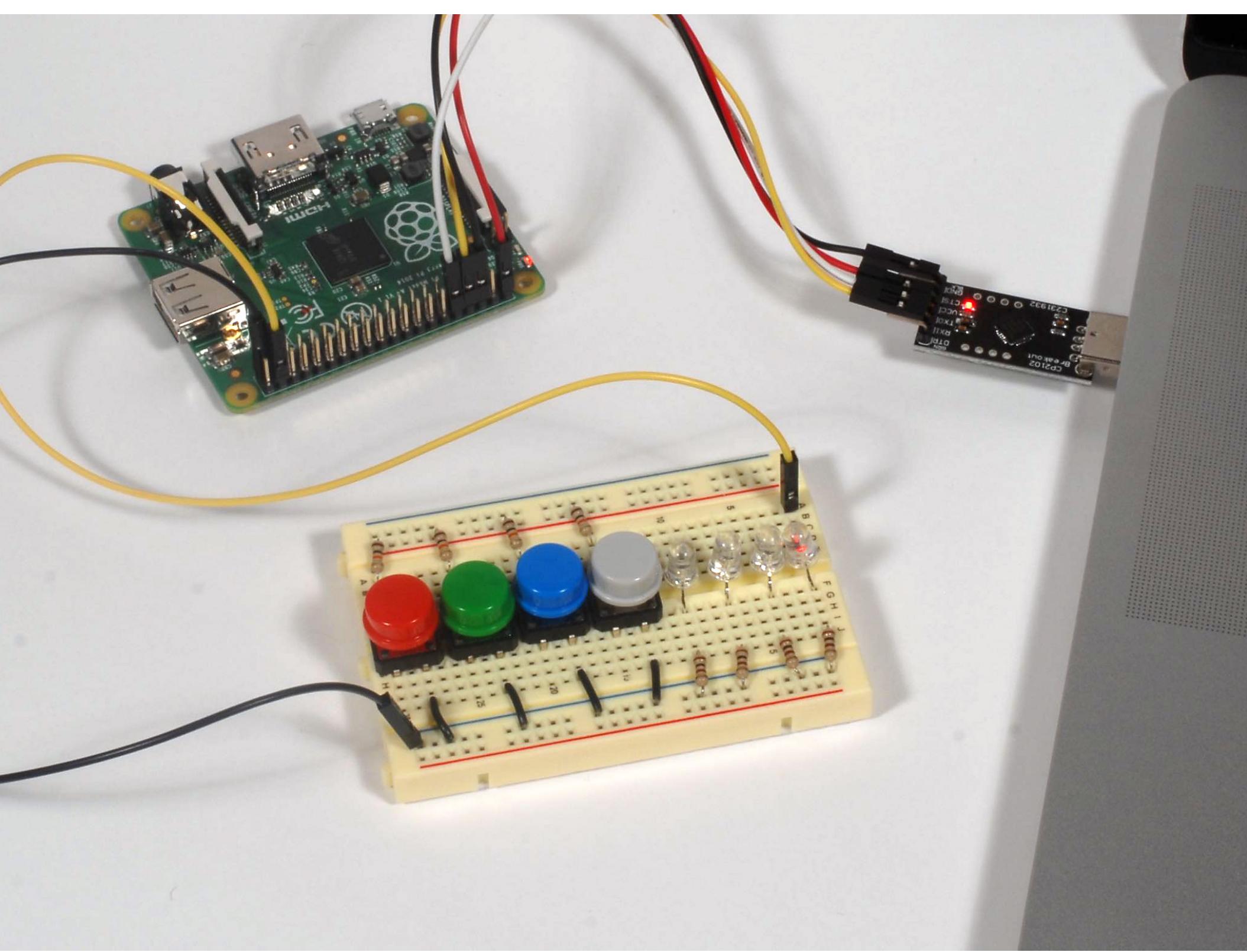
# **Package on Package**

**Broadcom 2865 ARM Processor  
ARM1176JZF-S 700 MHz  
ARMv6 Architecture**



**Samsung 512MB SDRAM**







# **Policy**

**Lab fee of \$50 (collected during 1st lab)**

**You break/lose it, you replace it**

# **Course Topics**

**cs107e.github.io**

# **§1 Baremetal Programming**

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

# **§2 Personal Computer**

**1. Keyboard**

**2. Graphics**

**3. Interrupts**

***Guest Lecture***

# **§3 Additional Topics**

- 1. Sensors**
- 2. Signed and unsigned arithmetic**
- 3. Towards Linux and beyond**

# **Administration**

# **Weekly Cadence**

**Each week has a major topic**

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

**Mandatory lab on Tue or Wed evening from 7-9 pm in Gates 325**

**Assignment due following Tue at 7 pm (before Tue lab)**

# **Laboratories**

**Gates 325**

**Attendance is mandatory**

**Hands-on exercises**

**Submit completed check-list**

**Leave ready to do assignment**

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

**Lab: access to tools and supplies**

# **Assignments**

**7 assignments**

- Build on each other**

**Two parts**

- Basic**
- Extensions**

**Final project**

**NO EXAMS**

# **Grading**

**Basic - 5 points**

**Extension - 5 points**

**Fully working system at the end - 5 points**

**A ~= 7 fully functional basic assignments + 3 extensions  
+ outstanding final project**

**B ~= 7 functional basic assignments + good final project**

# **First Week**

# **Questionnaire**

**Will send accept by Tue**

# **Assignment 0**

**Subscribe to cs107e in piazza**

**Attend cs107 UNIX labs**

**Assignment 0**

- Using git and github**
- Submit your lab preference**

**Read and understand basic guides**

# **Basic Electricity**

**Voltage and current**

**Ohms Law :  $V = I R$**

**Power :  $P = I V$**

**Driving an LED**

**Transistor switches**

**Breadboarding**

**Guide: [electricity.md](#)**

# **Number Representations**

**Binary representation**

**Hexadecimal**

**Bit operators**

**Guide: number.md**

# **Unix Command Line**

**Moving around the file system**

**Creating, moving, and deleting files**

**Compiling and running programs**

**Profiles and paths**

**Guide: unix.md**

**Note: Attend cs107 labs this week**