Embedded Systems (2)

- Will start at 15:10
- PDF of this slide is available via ScombZ

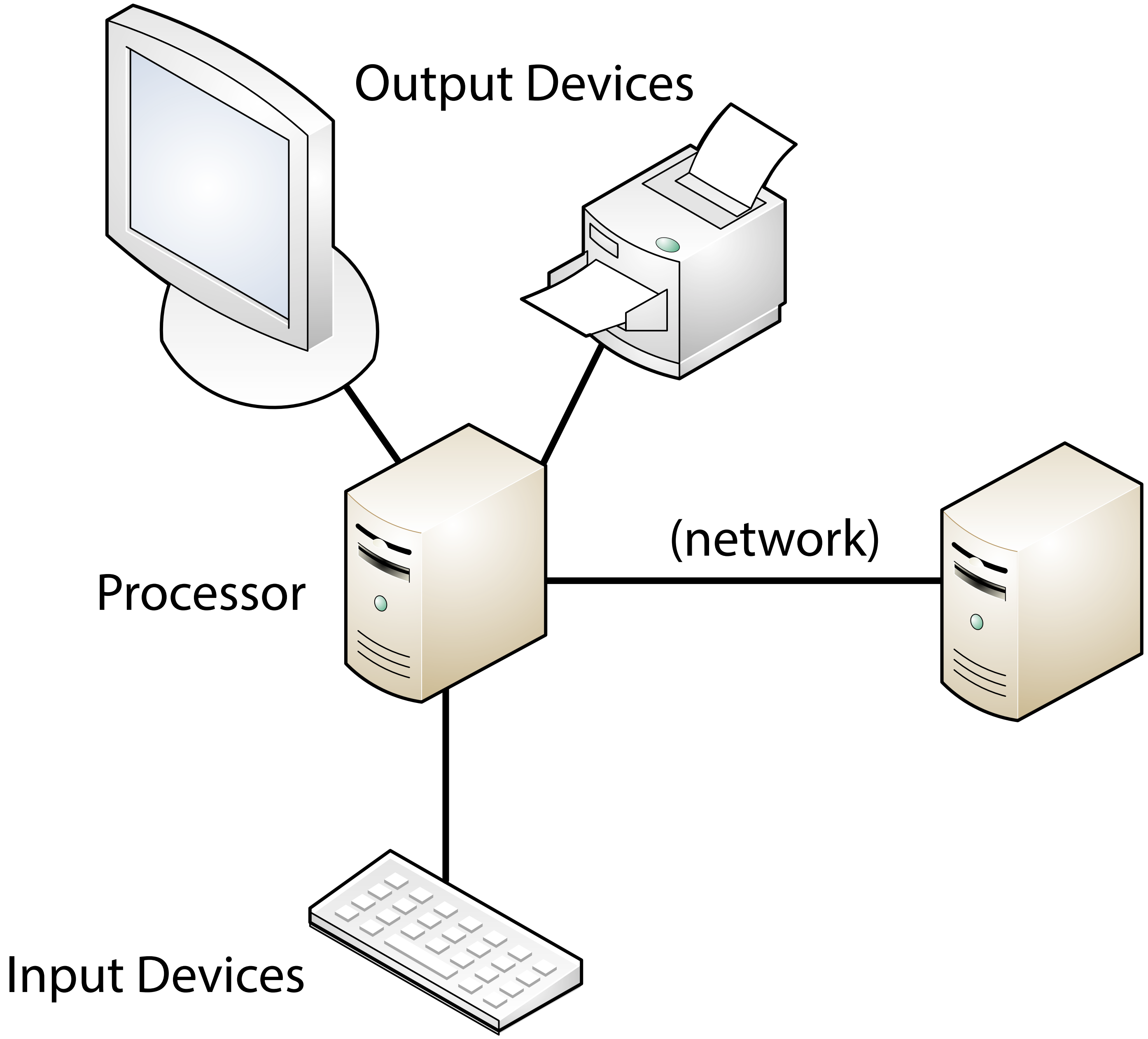
Hiroki Sato <i048219@shibaura-it.ac.jp>

15:10-16:50 on Wednesday

Targets At a Glance

- **What you will learn today: Hardware Architecture (1)**
  - A small-scale embedded system using a processor
    - Structure as a computer system
    - What internals of a processor look like
    - How your program works
  - The first project: simple use of GPIO and LED (1)
    - Illuminations
    - LED and basics of electric circuits

Computer System



Output Devices

Processor

(network)

Input Devices

Simplified model of information processing flow

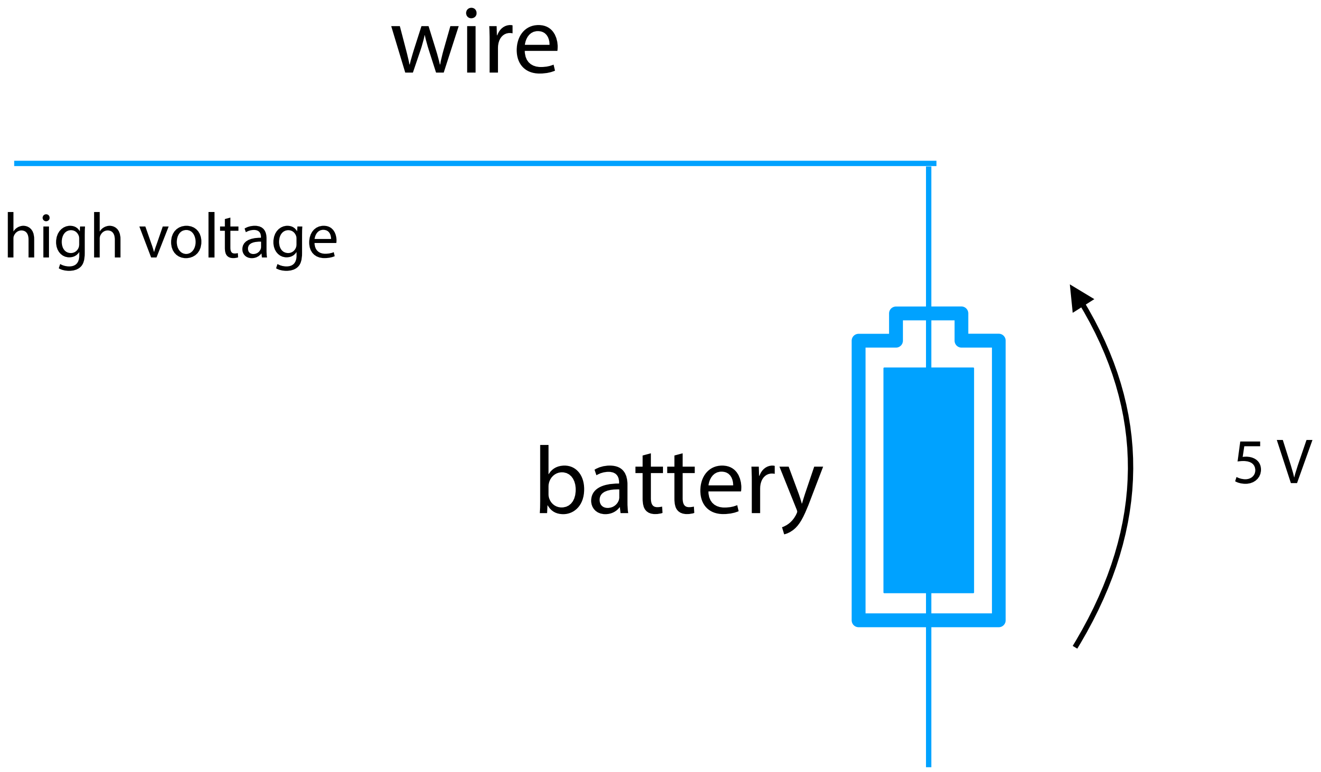
Hardware Architecture

- **Stored-Program Computer**
  - **CPU**: central processing unit, or processor
    - reads **instructions (program)** and executes them
  - **Memory**
    - stores **programs** and **data**
  - **Input and output interfaces**
    - to communicate with **other devices**

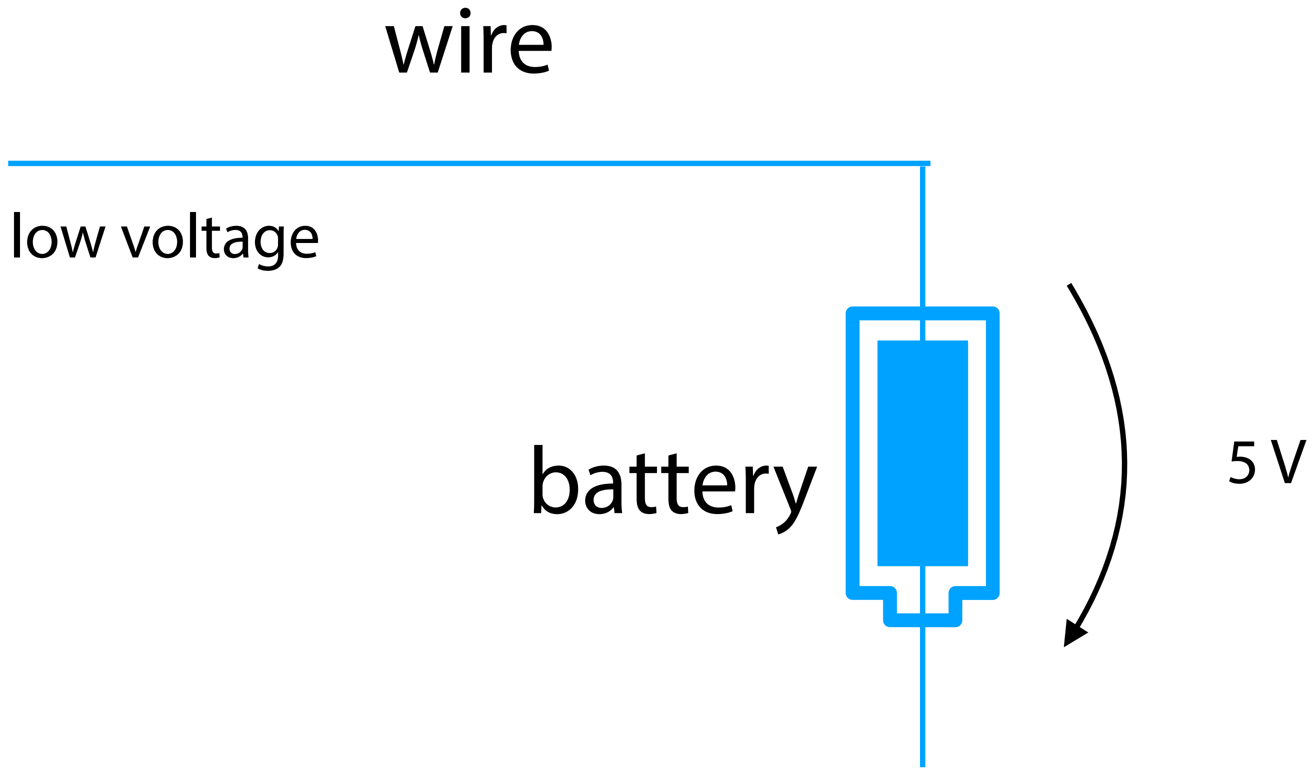
Hardware Architecture

wire

Hardware Architecture



Hardware Architecture



Hardware Architecture

A wire

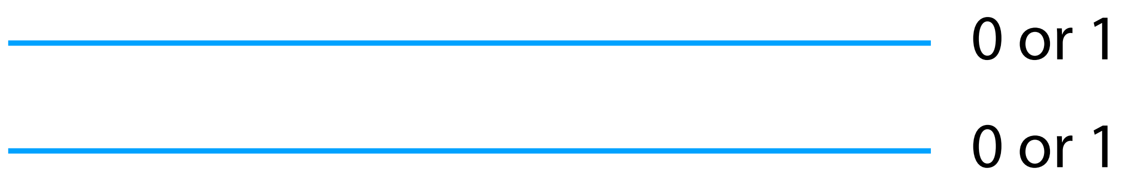
can be high voltage or low voltage

→ a single wire can deliver information

high voltage means "1"
low voltage means "0"

Hardware Architecture

Two wires

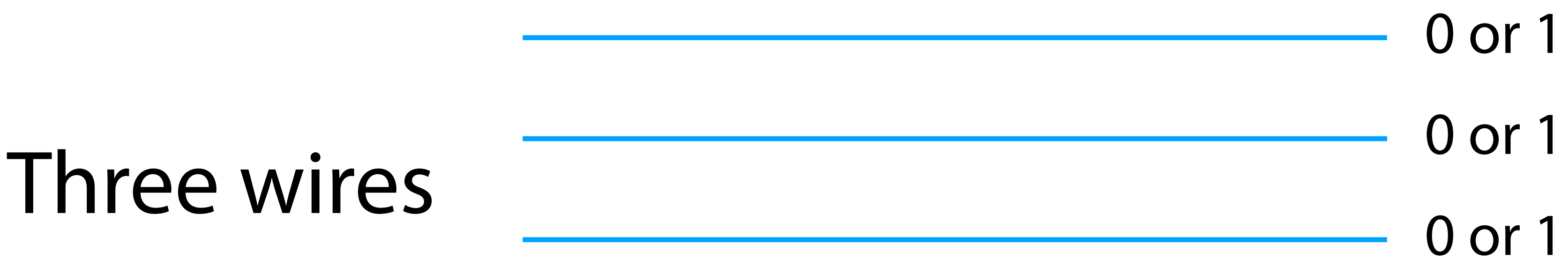


0 or 1

0 or 1

binary numeral	00	01	10	11
decimal numeral	0	1	2	3

Hardware Architecture



Three wires

0 or 1

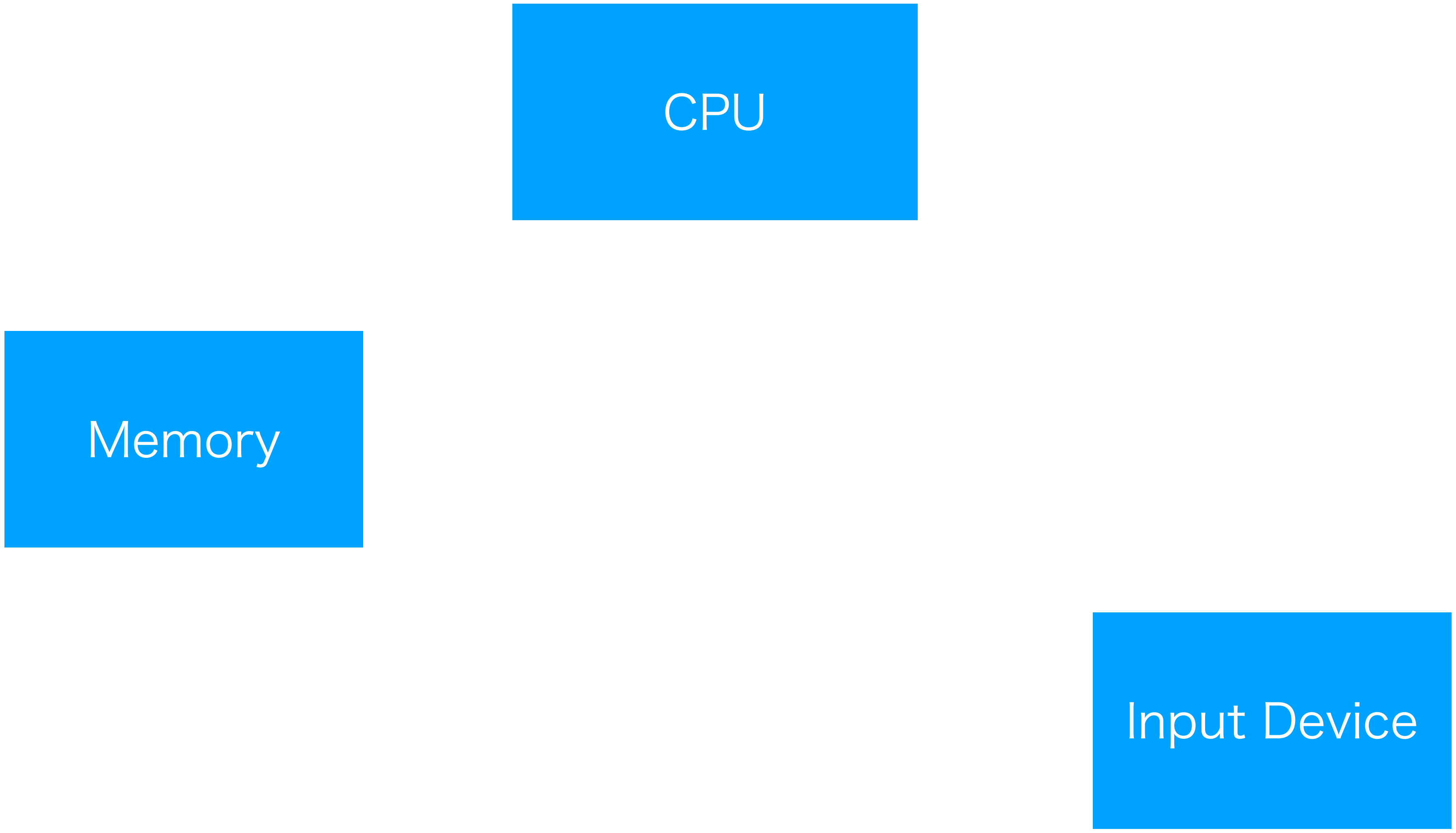
0 or 1

0 or 1

00	01	10	11	100	101	110	111
0	1	2	3	4	5	6	7

Three wires can count from 0 to 2^3 - 1

Hardware Architecture

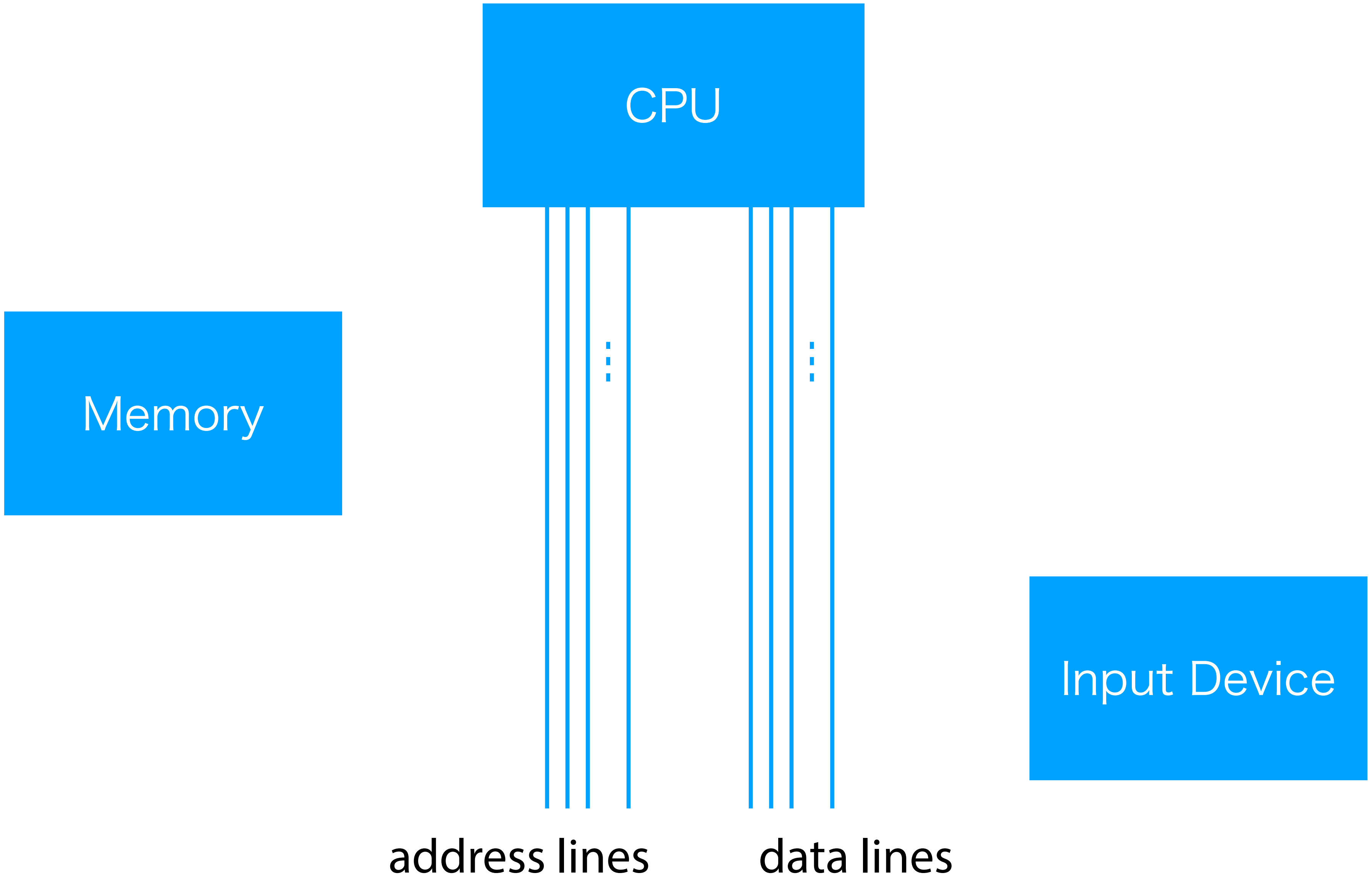


CPU

Memory

Input Device

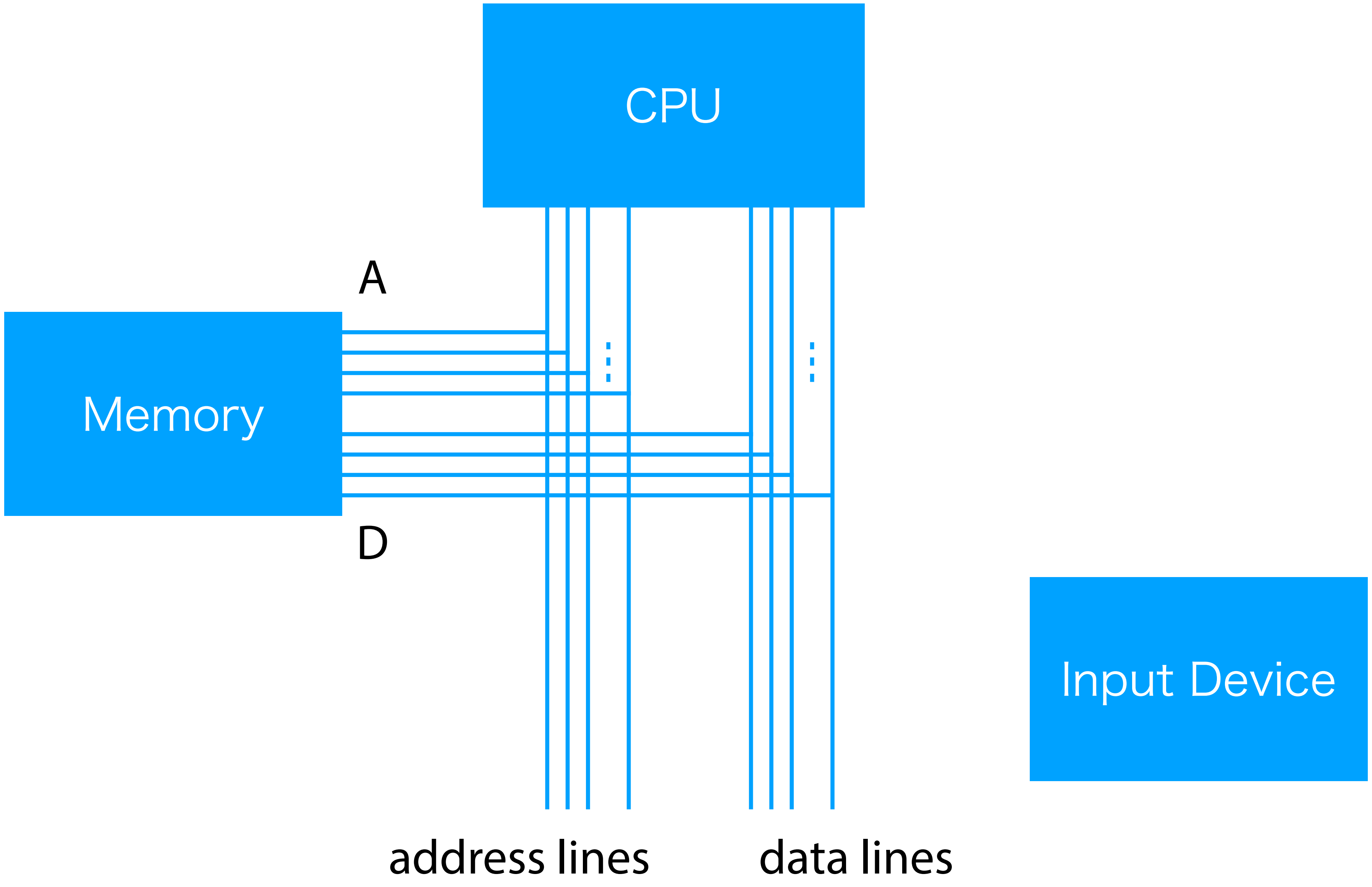
Hardware Architecture



what device wants to communicate

data themselves

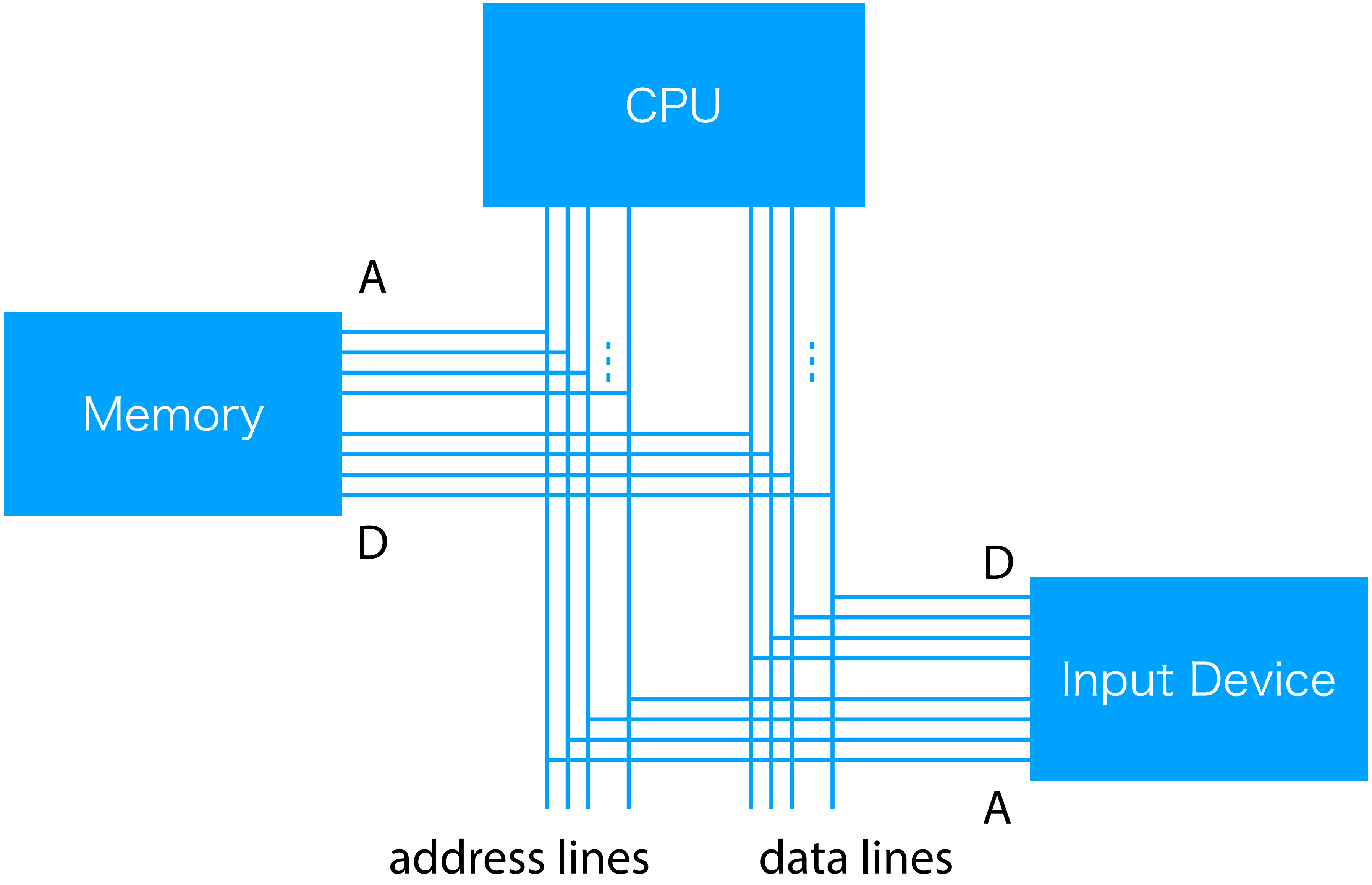
Hardware Architecture



what device wants to communicate

data themselves

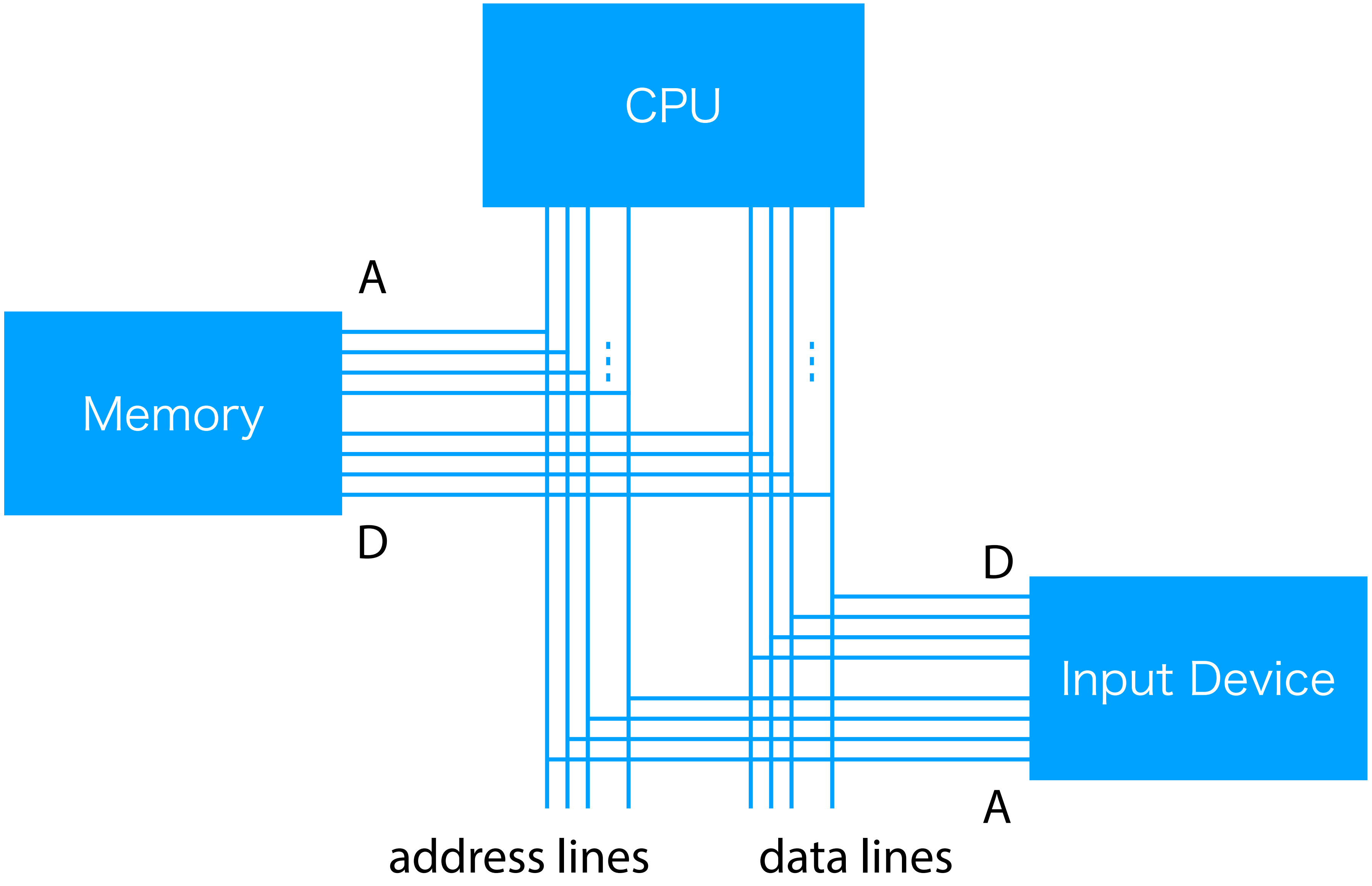
Hardware Architecture



what device wants to communicate

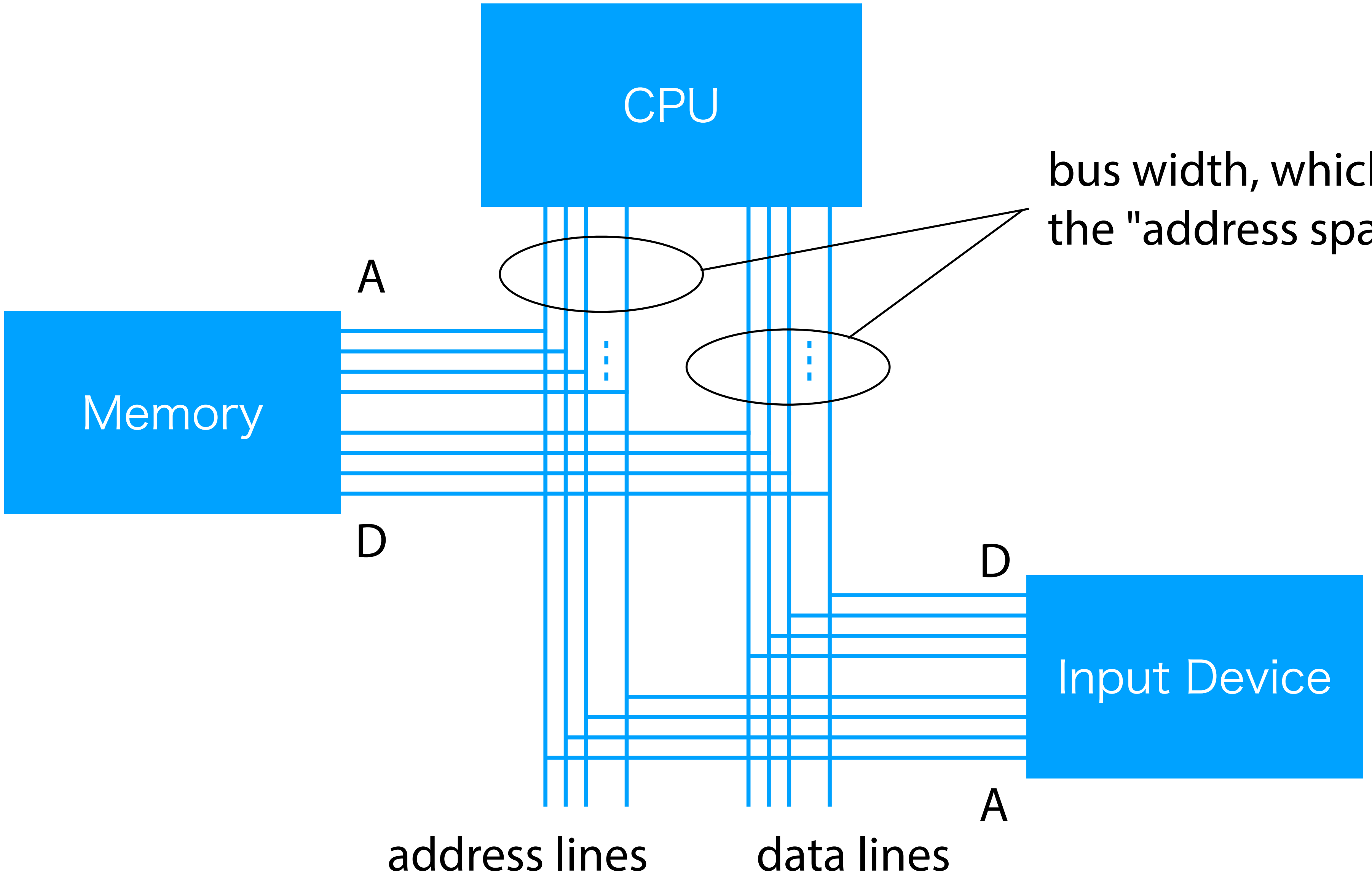
data themselves

Hardware Architecture



"bus" topology

Hardware Architecture



CPU

Memory

Input Device

A

D

D

A

address lines

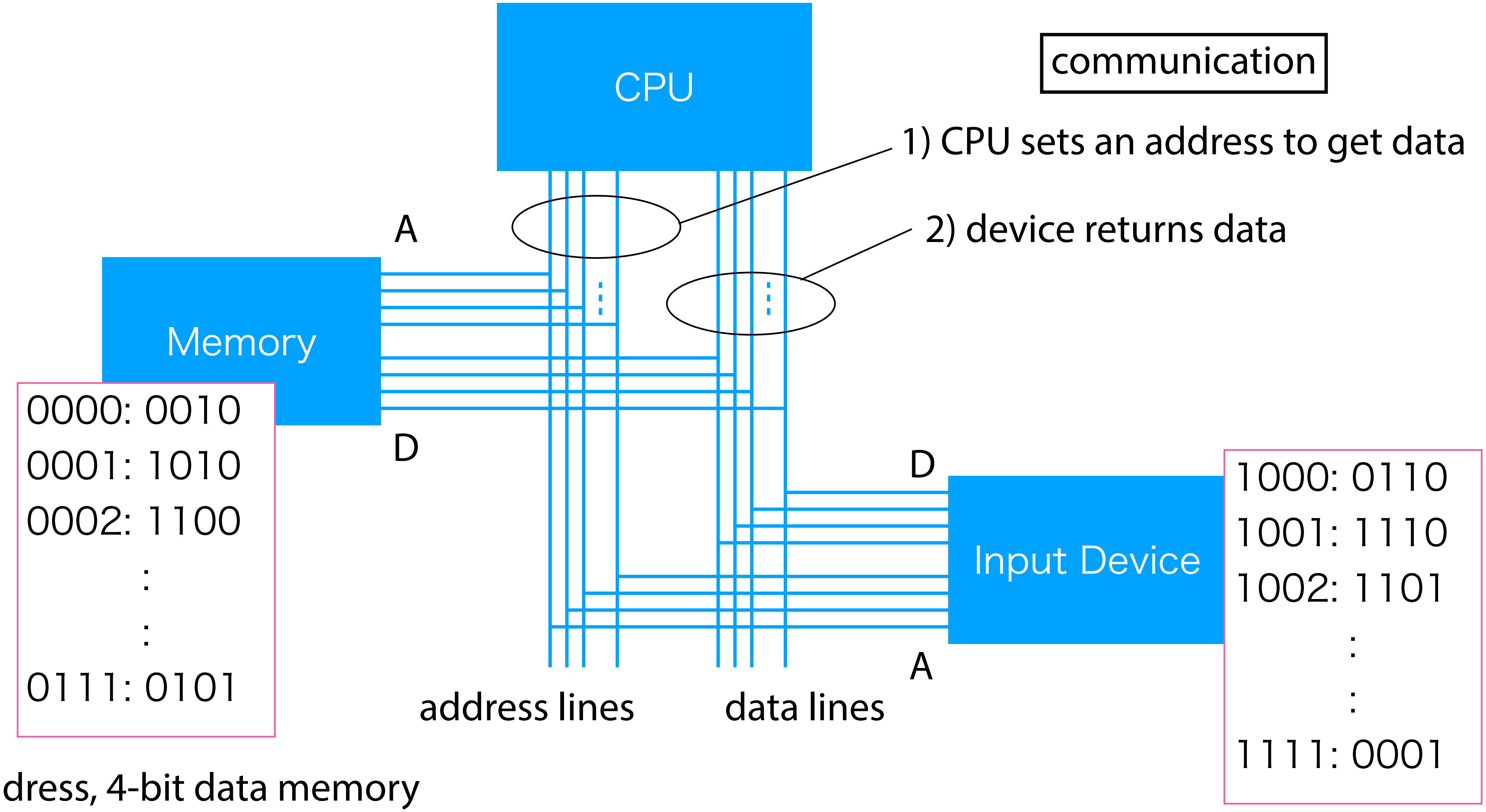
data lines

bus width, which determines the "address space"

"bus" topology

Hardware Architecture

4-bit address, 4-bit data processor



communication

1) CPU sets an address to get data

2) device returns data

0000: 0010
0001: 1010
0002: 1100
:
:
0111: 0101

1000: 0110
1001: 1110
1002: 1101
:
:
1111: 0001

3-bit address, 4-bit data memory

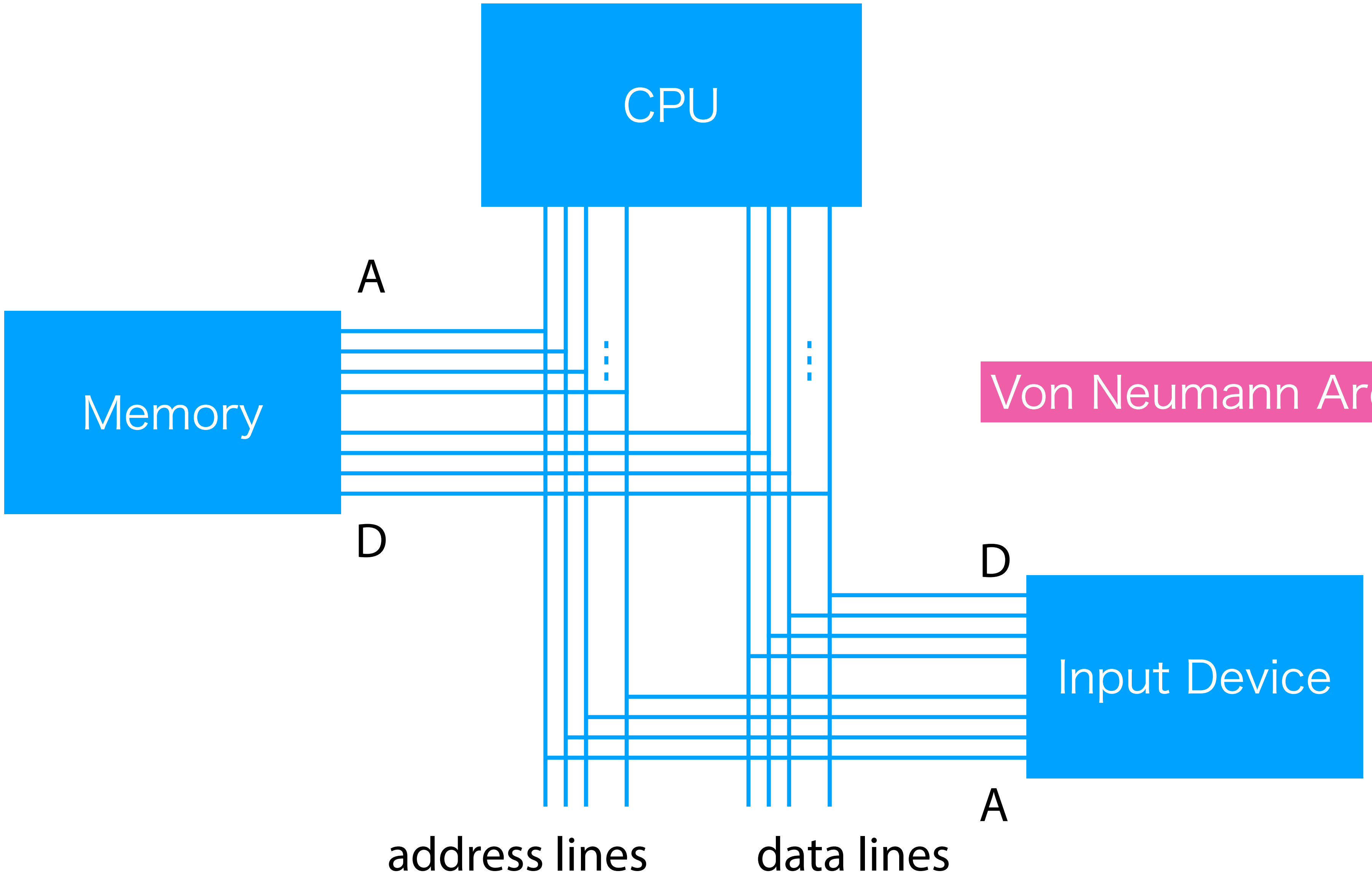
3-bit address, 4-bit data device

Development Platform

- **Arduino**: a vendor of single board microcontroller kits.
- **Arduino Uno**: a product of Arduino.
    - ATmega328P 8-bit processor, 16MHz
    - 32kB Flash, 2kB SRAM, 1 EEPROM
    - GPIO 20pin, ADC 4ch

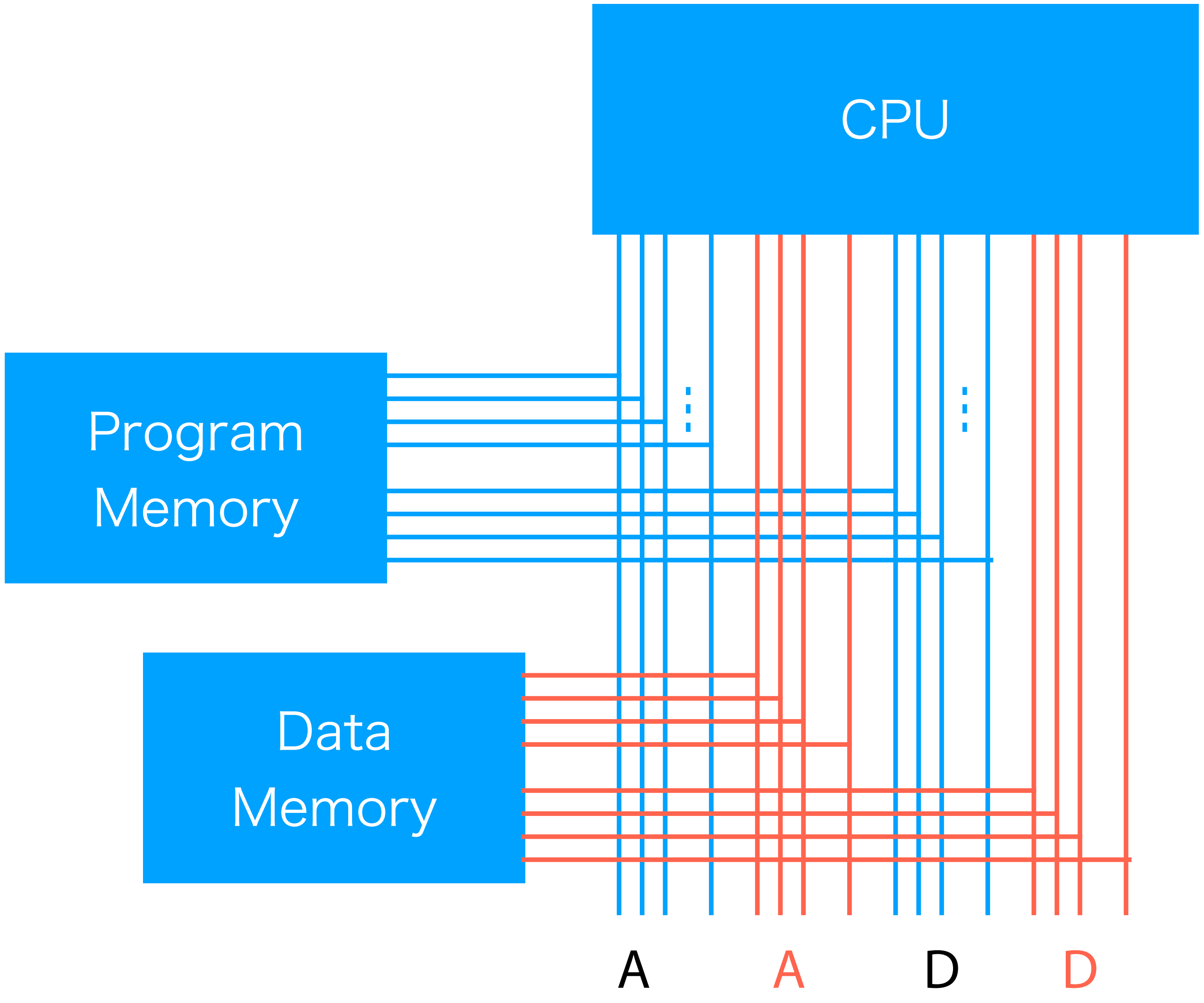
8-bit data bus
  = 2^8 = 256 (1 byte)
11-bit address bus for data memory
  = 2^11 = 2048 → 2kB
14-bit address bus for program memory
  = 2^14 = 16384 (16384 x 2 = 32kB)
Note: every instruction is 2 bytes wide.

Hardware Architecture



Von Neumann Architecture

Hardware Architecture

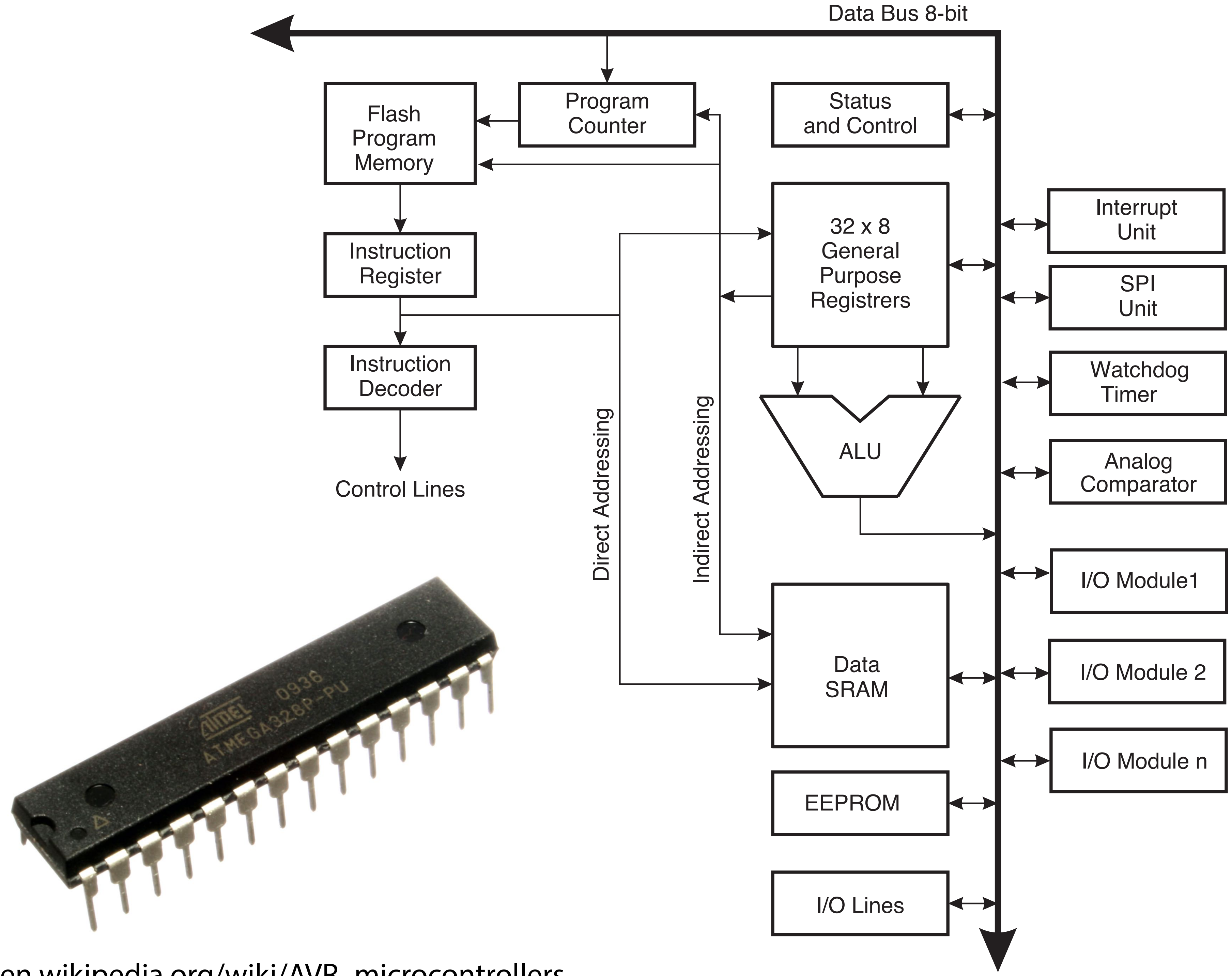


Harvard Architecture

Hardware Architecture

- **Von Neumann Architecture**
  - shares the same bus for multiple devices
  - popular in general-purpose computers
  - **Pros:** simple
  - **Cons:** memory access speed can be a bottle-neck
- **Harvard Architecture**
  - has buses for each device (especially memory for program and data)
  - **Pros:** faster than Von Neumann
  - **Cons:** more wires on the buses and complexity of instruction set
- **ATmega328P adopts a modified Harvard architecture.**

ATmega328P



Data Bus 8-bit

Flash Program Memory

Program Counter

Status and Control

Instruction Register

32 x 8 General Purpose Registrers

Interrupt Unit

SPI Unit

Instruction Decoder

Watchdog Timer

Control Lines

Direct Addressing

Indirect Addressing

ALU

Analog Comparator

I/O Module1

I/O Module 2

I/O Module n

Data SRAM

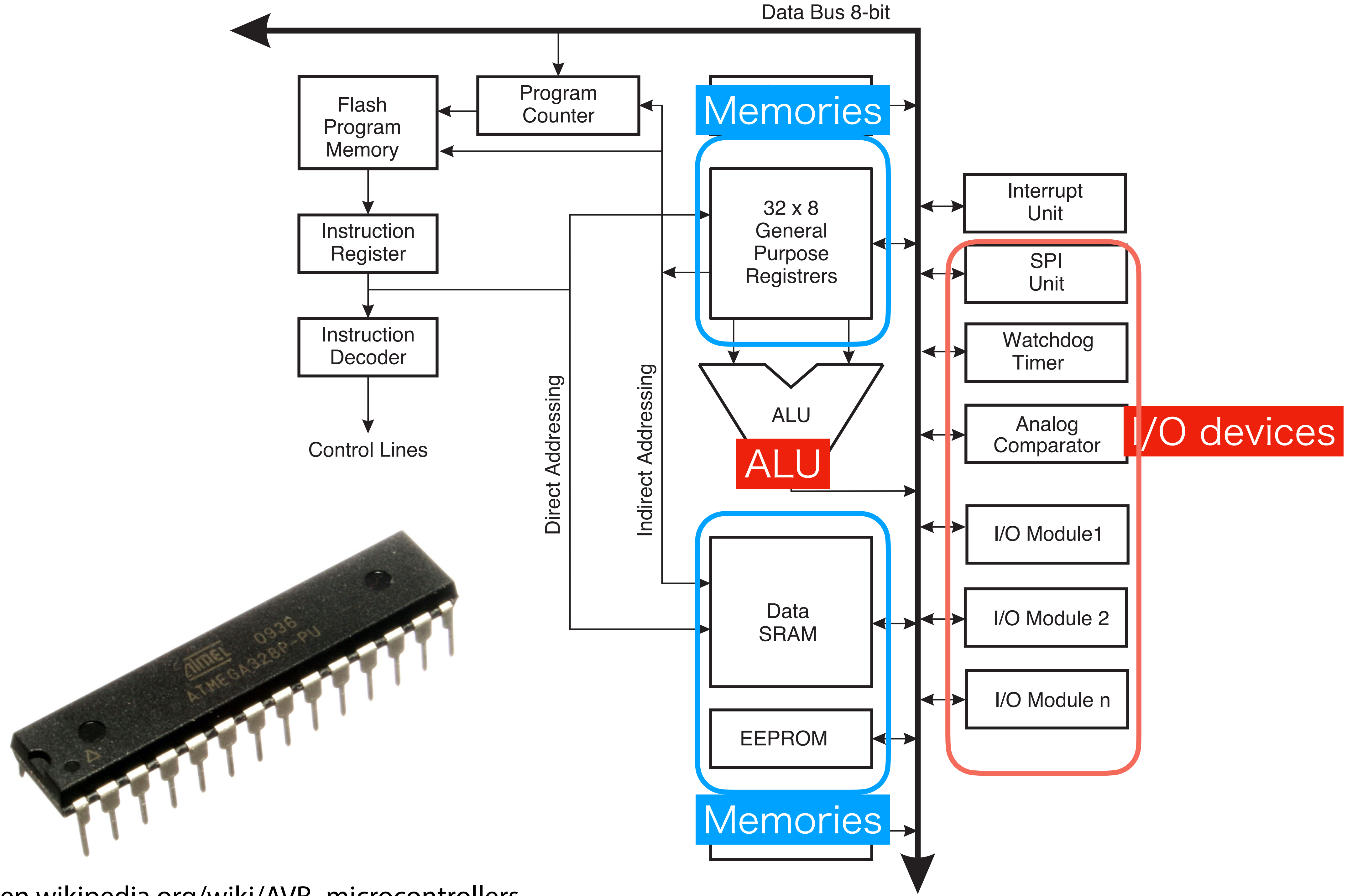
EEPROM

I/O Lines

https://en.wikipedia.org/wiki/AVR_microcontrollers

http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061B.pdf

ATmega328P



https://en.wikipedia.org/wiki/AVR_microcontrollers

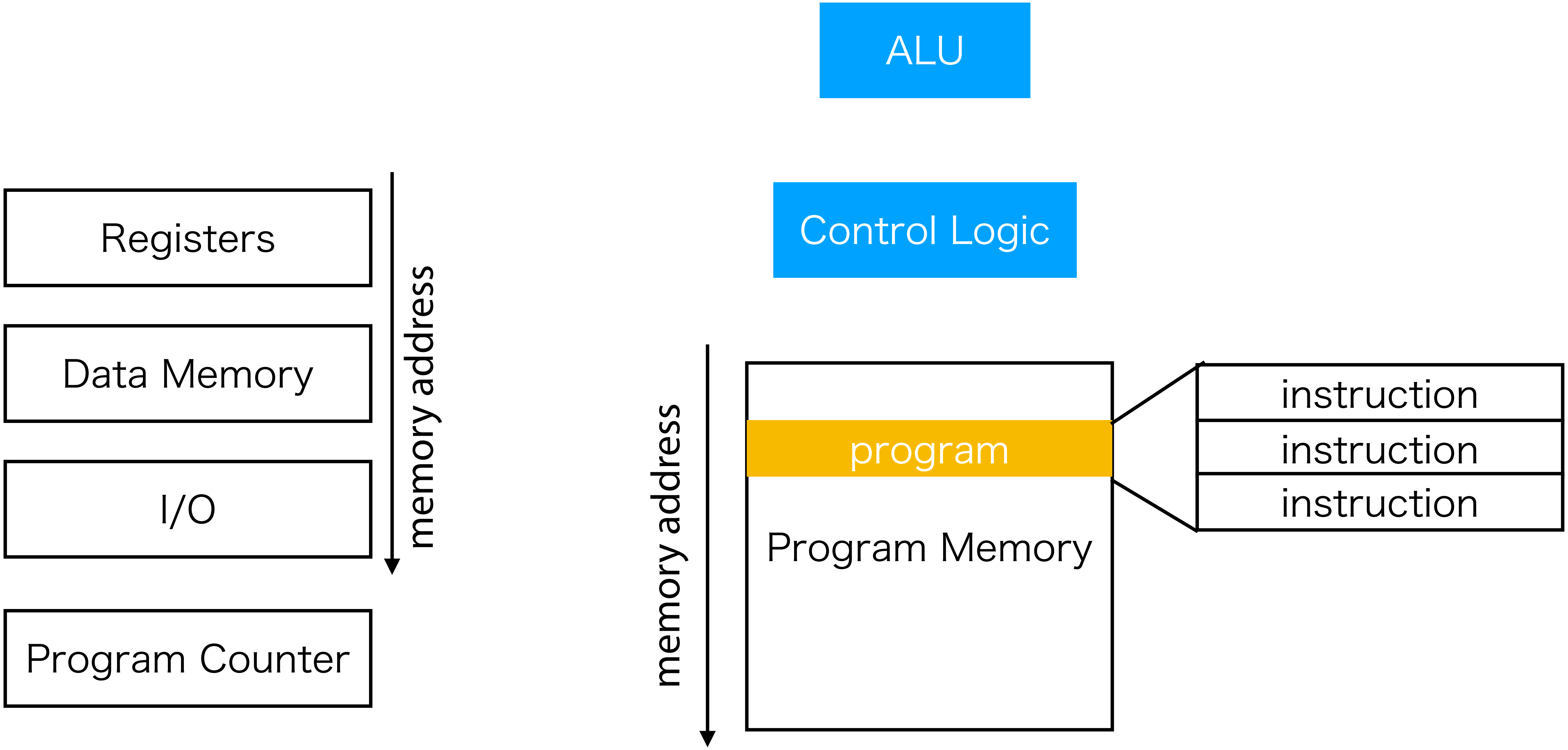
http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061B.pdf

Structure of Processor

- **ALU: Arithmetic Logic Unit**
  - Combinational digital circuit that performs arithmetic and bitwise operations
- **Registers: small memories inside the processor**
- **Control logic**
  - does bootstrapping and then puts the processor in the read-and-execute cycle.
  - Handles interrupts and I/O operations

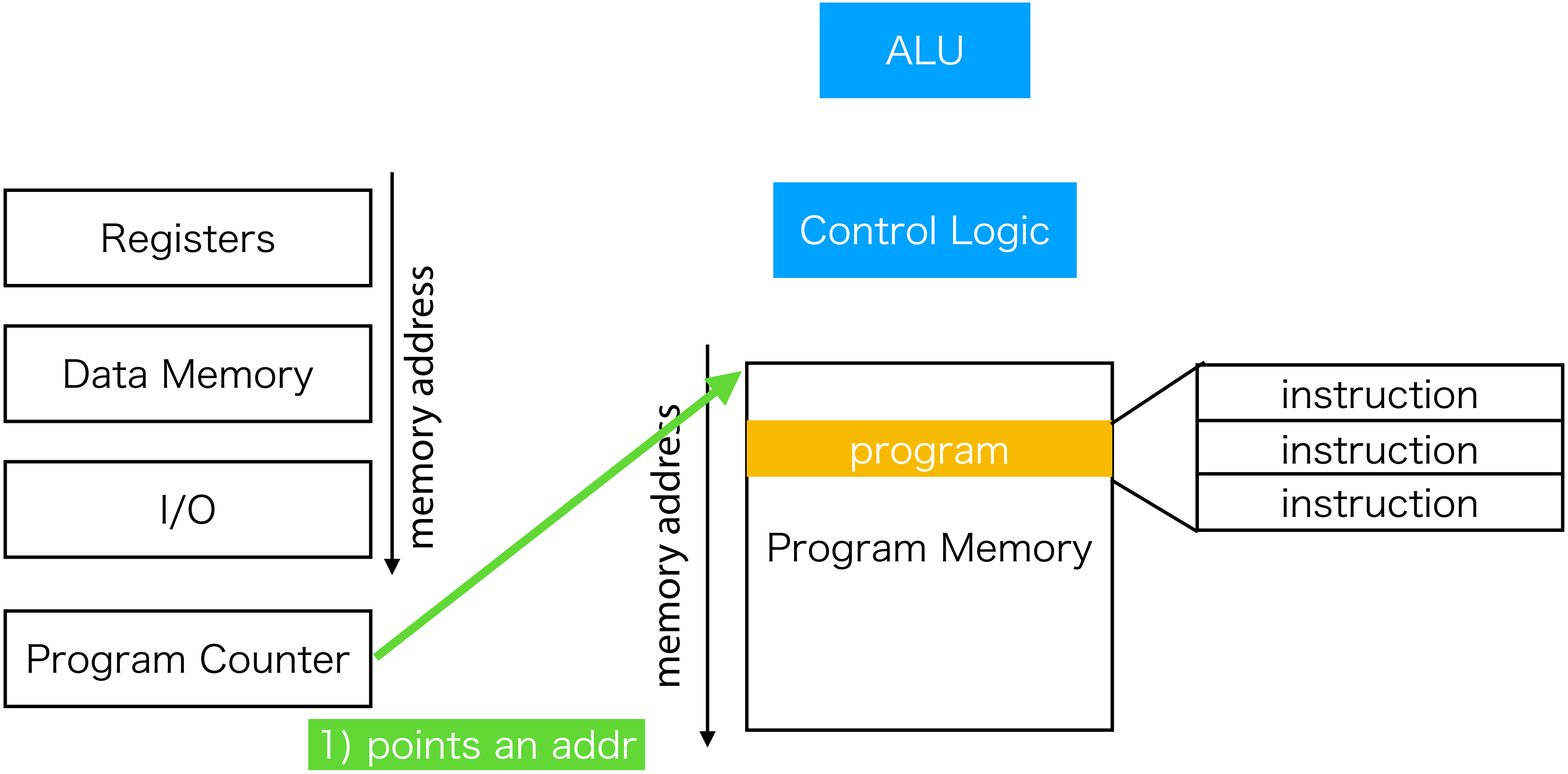
How Processor Works

- **Components of a processor from software perspective**



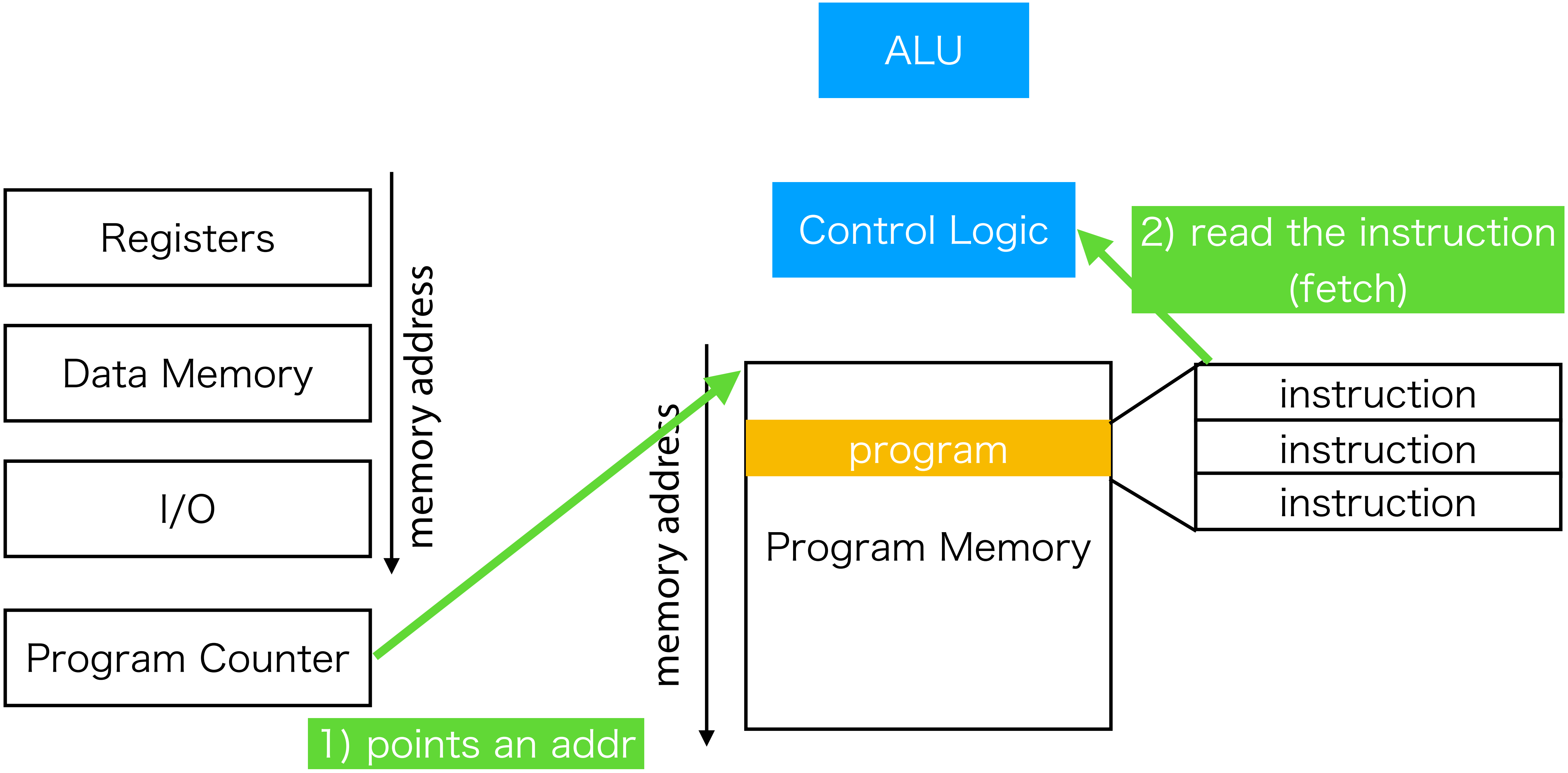
How Processor Works

- **Components of a processor from software perspective**



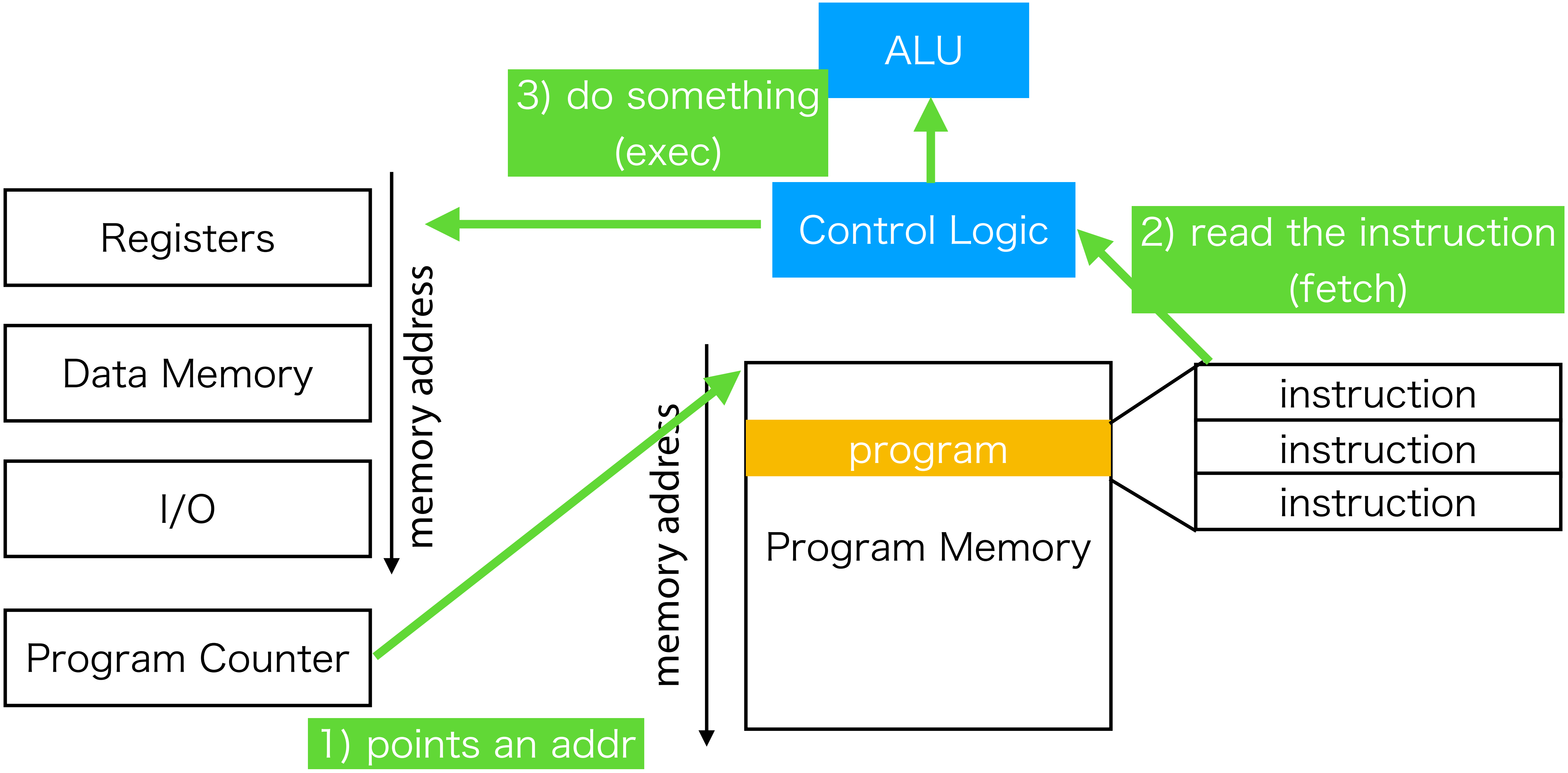
How Processor Works

- **Components of a processor from software perspective**



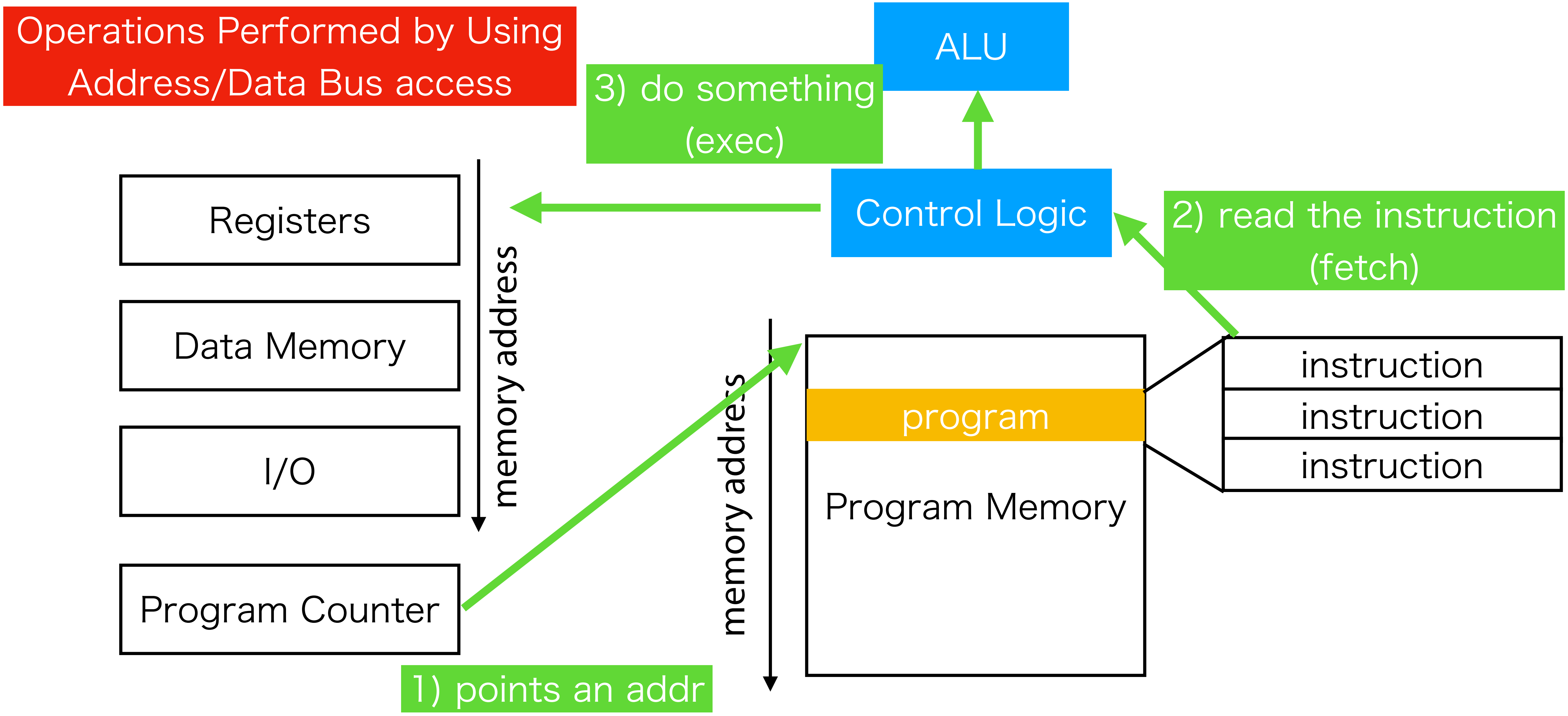
How Processor Works

- **Components of a processor from software perspective**



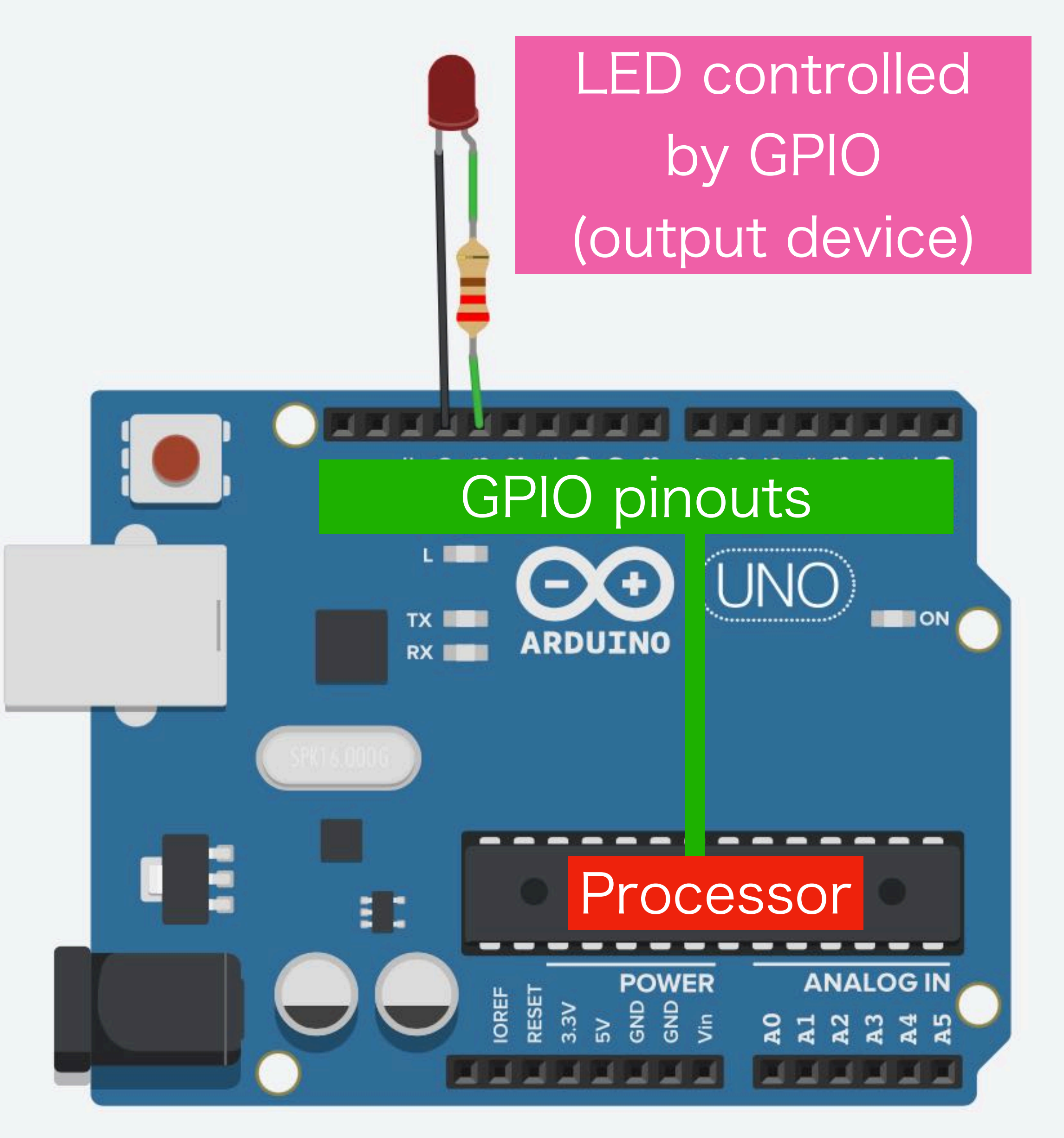
How Processor Works

- **Components of a processor from software perspective**

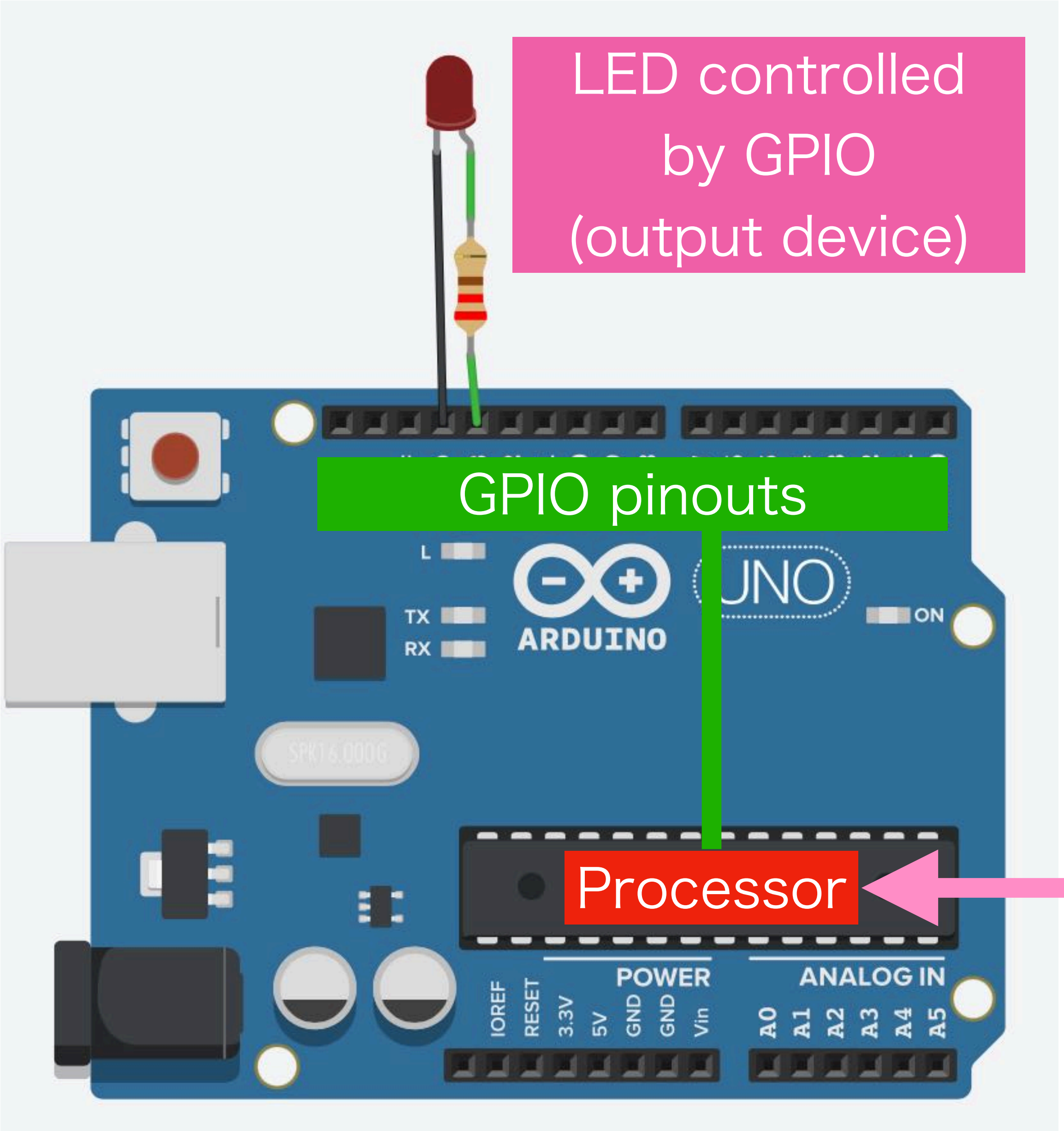


Input/Output Devices

- **GPIO**: General Purpose Input/ Output Interface
  - "Wires" connected to the address/data bus.
  - Can handle high and low voltages as 1 and 0



Putting them together



LED controlled by GPIO (output device)

GPIO pinouts

Processor

A program (you develop)

- This system has an LED as the output device only.
- You can still develop a program to control the output device.

Exercise

The First Project

- **Simple use of GPIO and LED (1)**
  - Illuminations
  - LED and basics of electric circuits

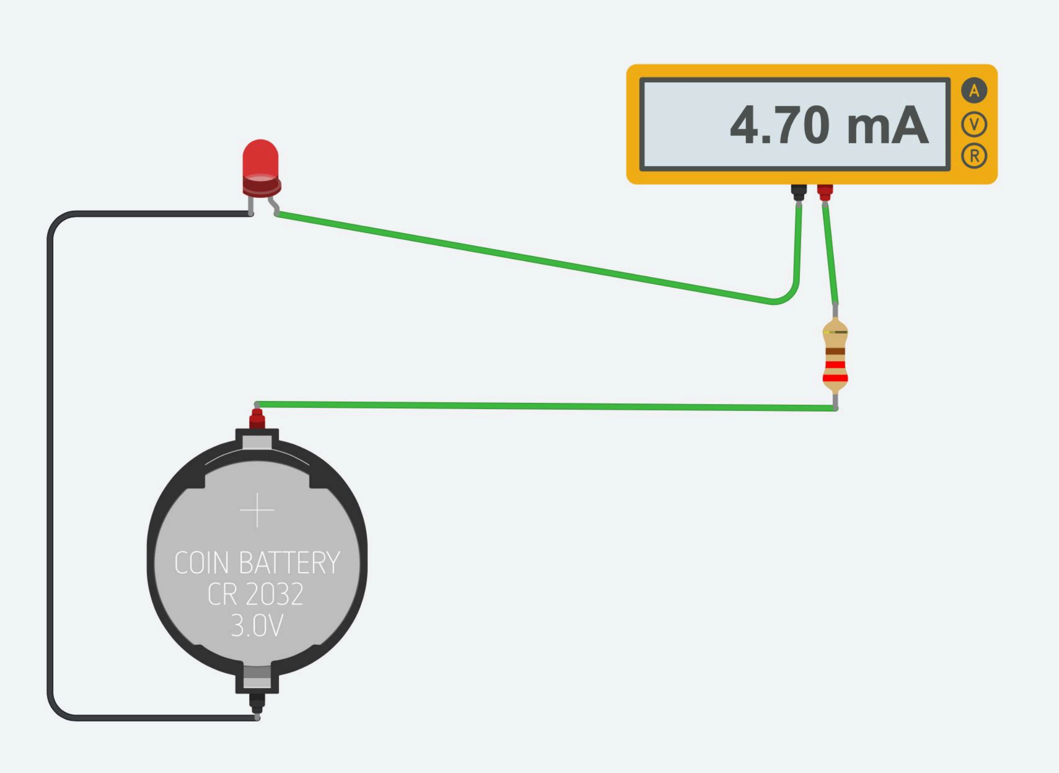
- **Visit TinkerCad**

  `https://www.tinkercad.com/joinclass/SKRUGE7BB`

- Enter your nickname. If you do not know it or could not access to the website, contact the teaching assistant

LED

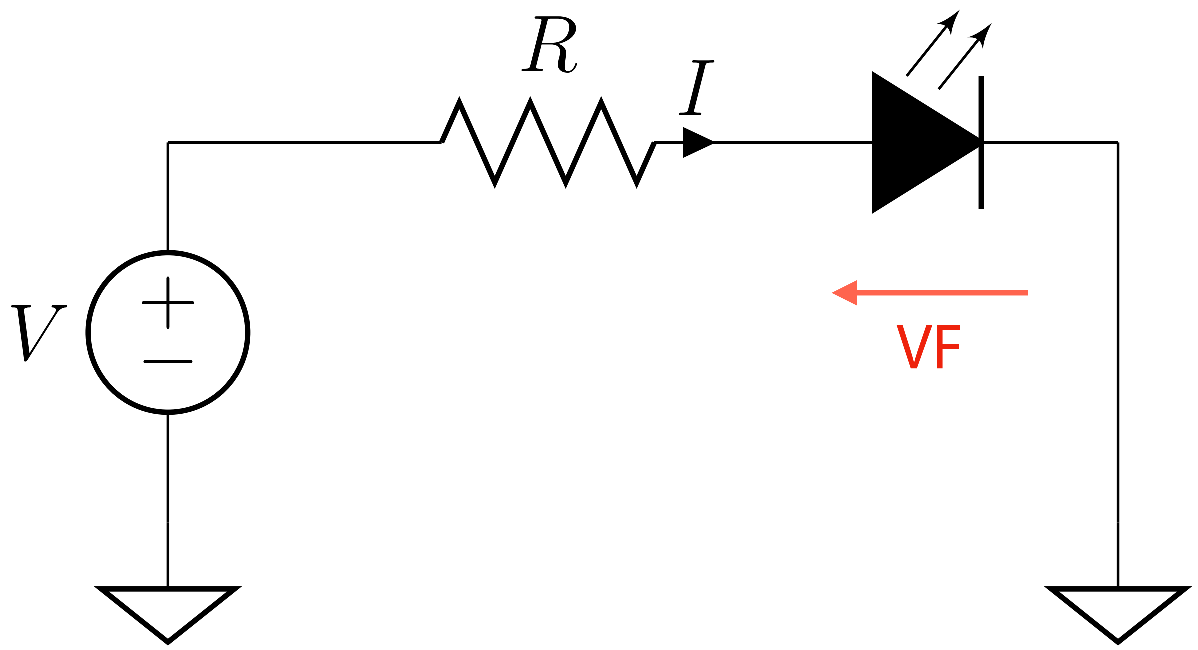
- **Light-emitting diode**
  - Two-terminal component.
  - Red: 2V, 10mA is typical.
  - More currents make it brighter, but >20mA breaks it.



https://en.wikipedia.org/wiki/Light-emitting_diode

LED

- **Light-emitting diode**
  - R is mandatory. This limits the current.
  - If R=0, the current can be high and it generates heat.



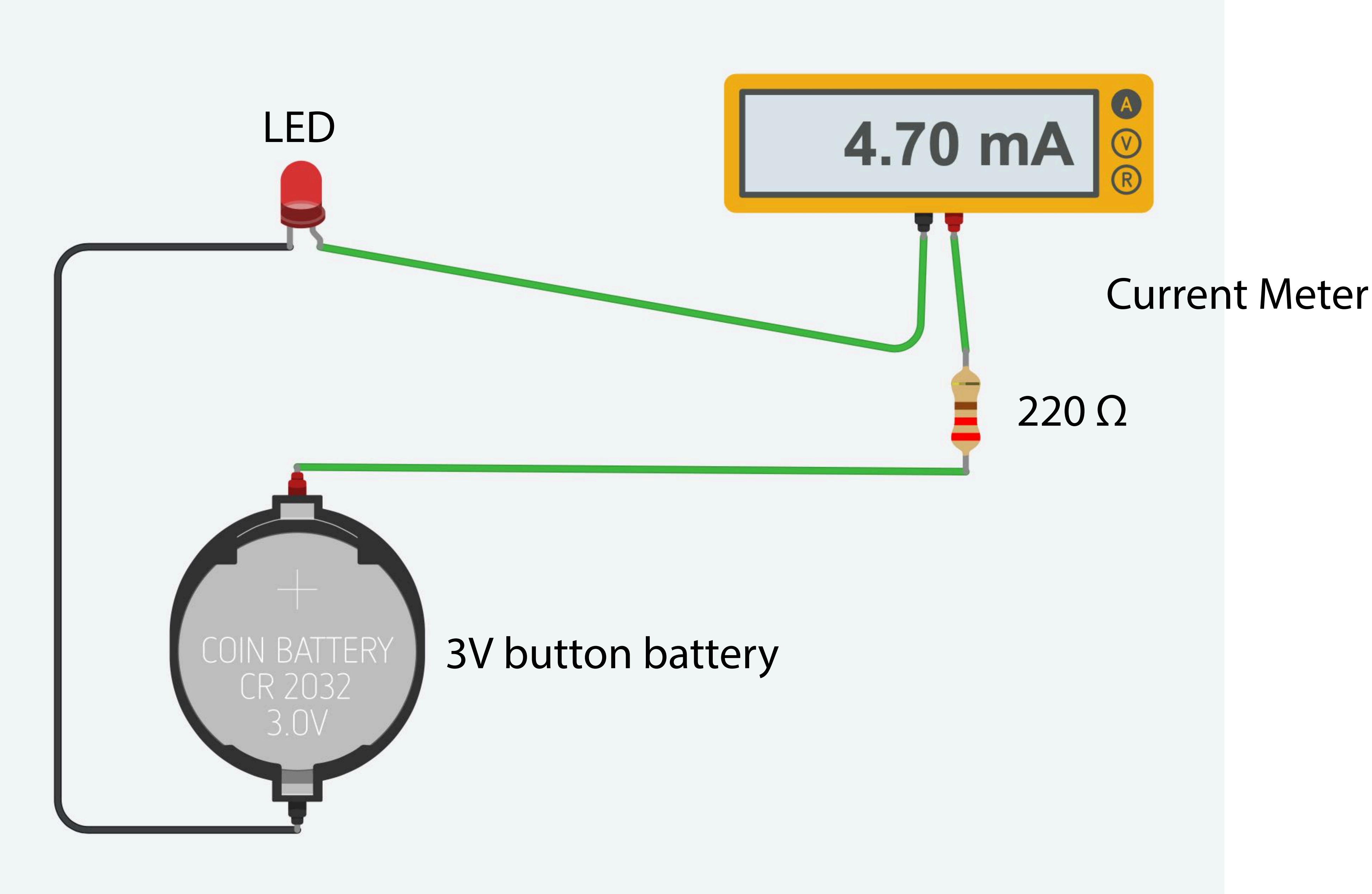
$$V = RI + V_F$$

$$\therefore R = \frac{V - V_F}{I}$$

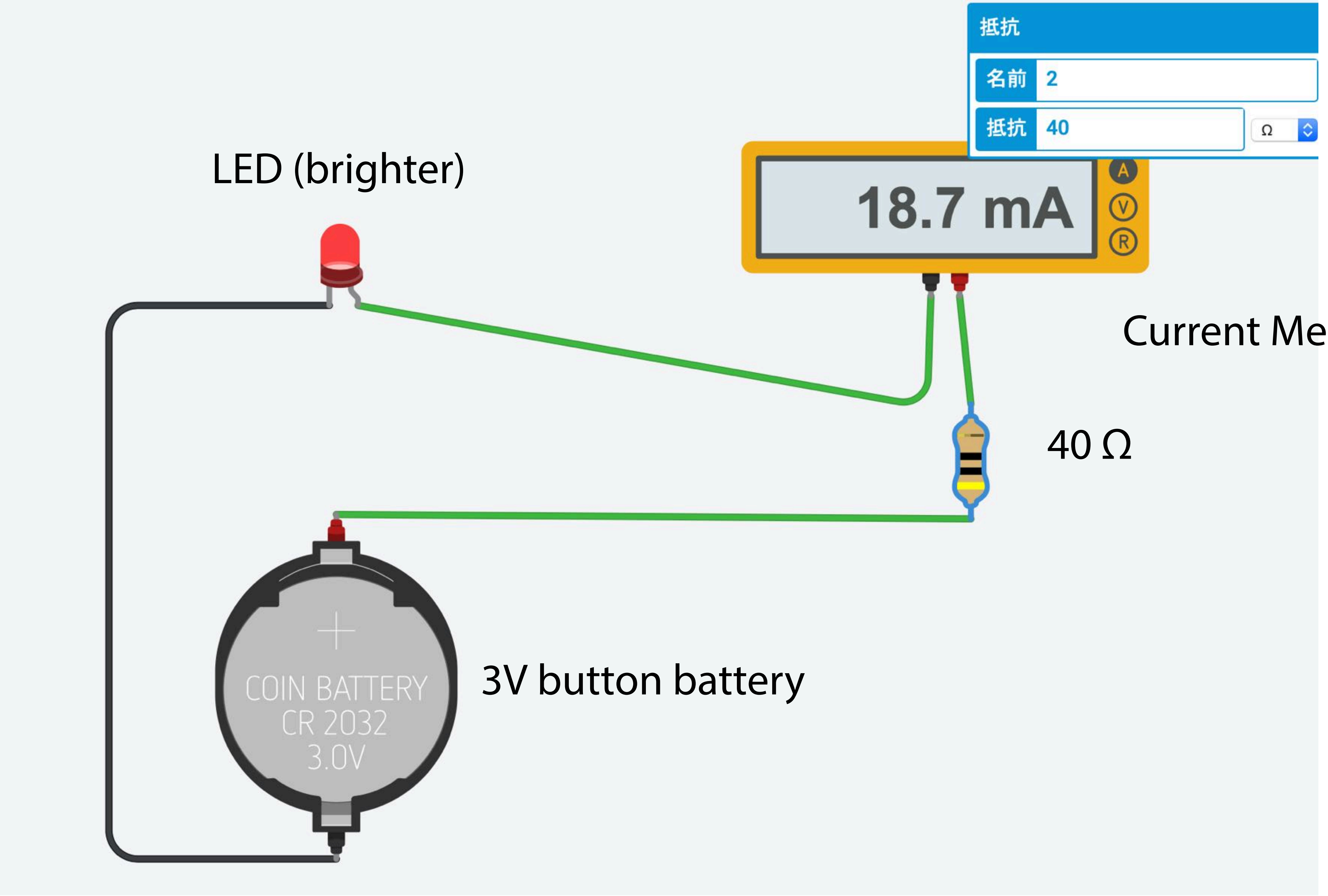
$$I = 10 \text{ mA}$$

$$V = 3 \text{ V}$$

Try it in Simulator



Try it in Simulator



抵抗

名前 2

抵抗 40 Ω

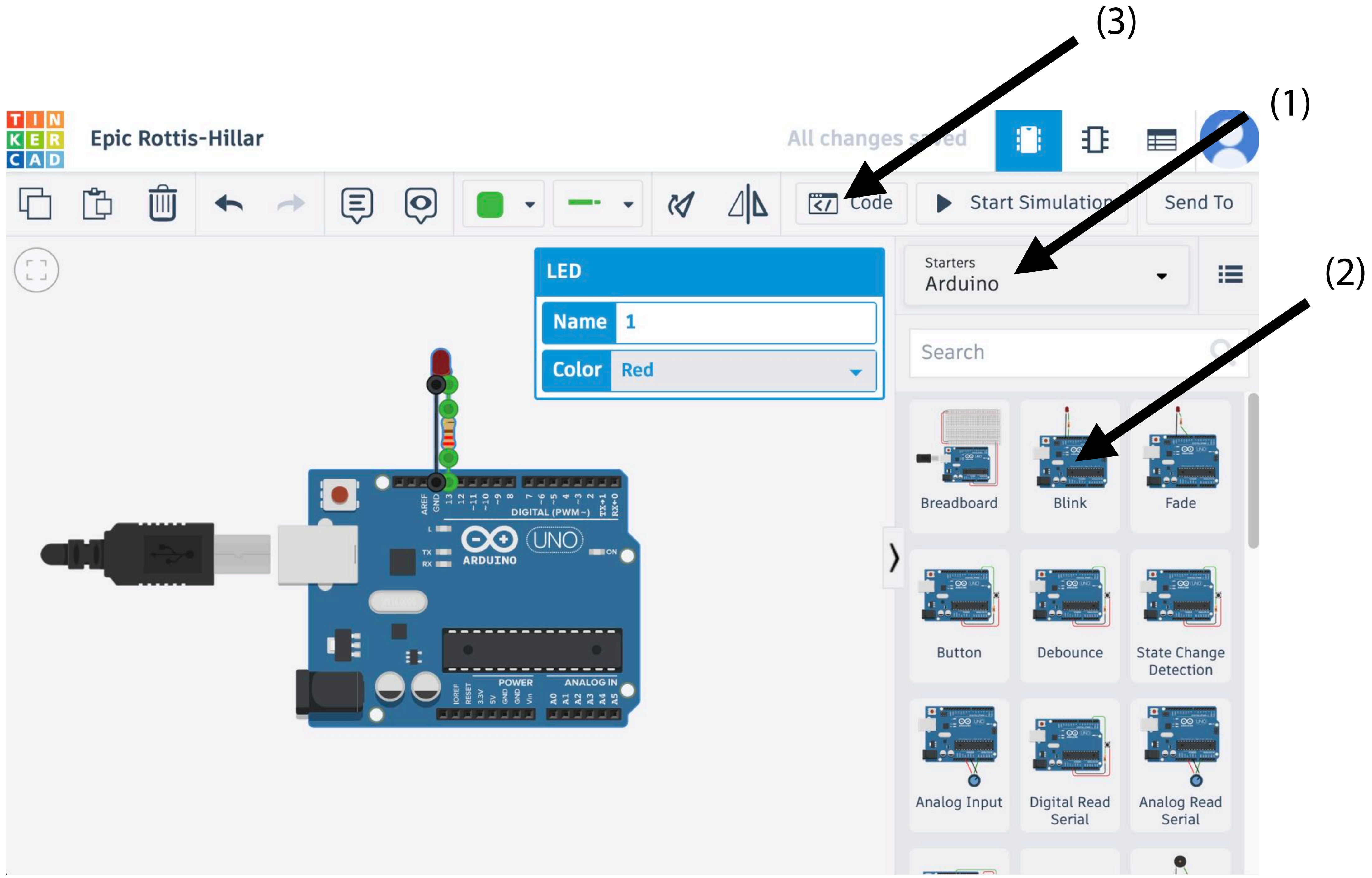
LED (brighter)

Current Meter

40 Ω

3V button battery

Try Blinker



LED

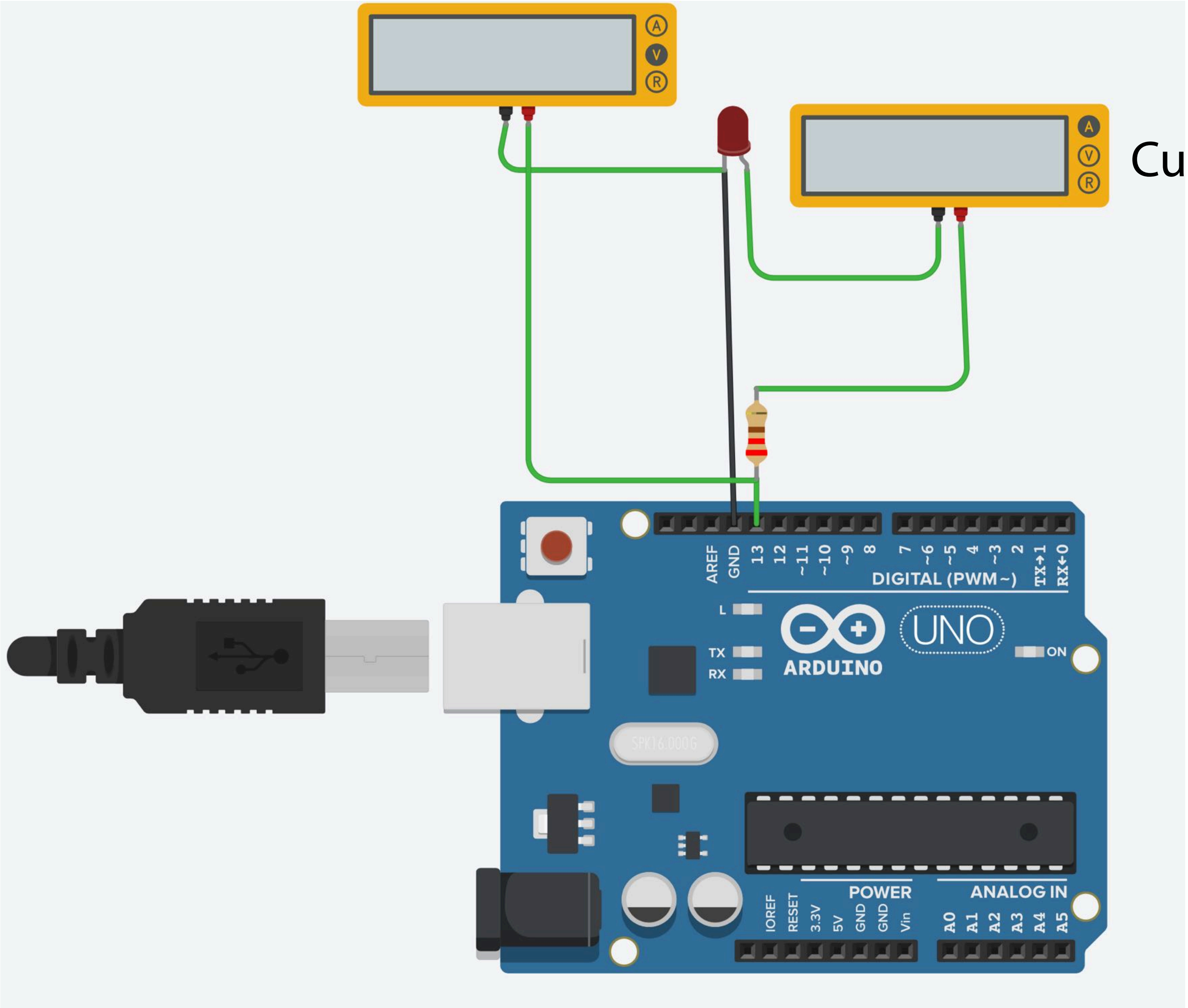
Name: 1

Color: Red

Starters: Arduino

Try Blinker

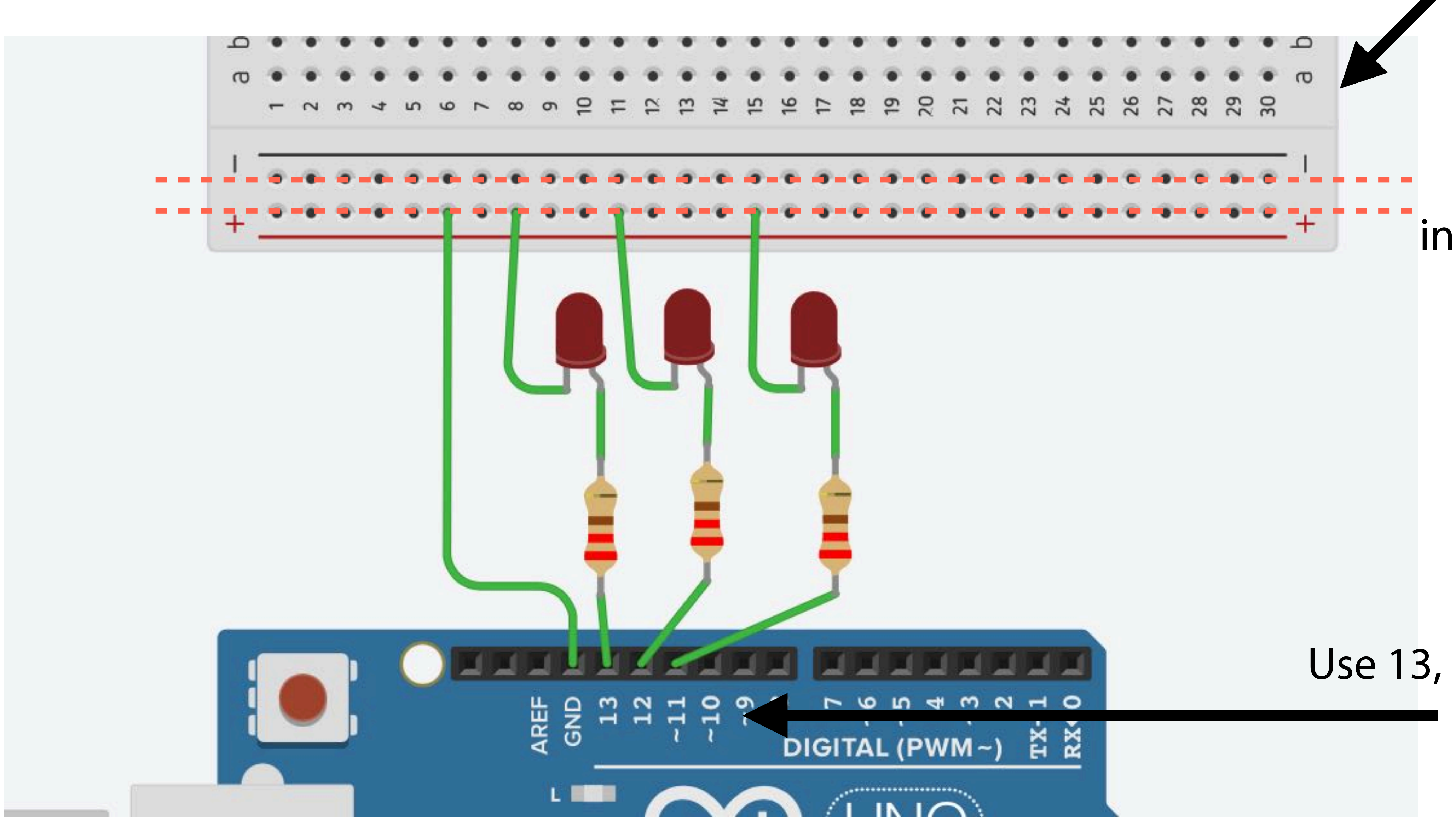
Voltage Meter



Current Meter

- **Questions:**
  - What voltage is supplied from the development board?
  - How do you determine the R value?

Try 3-LED Blinker



"breadboard"

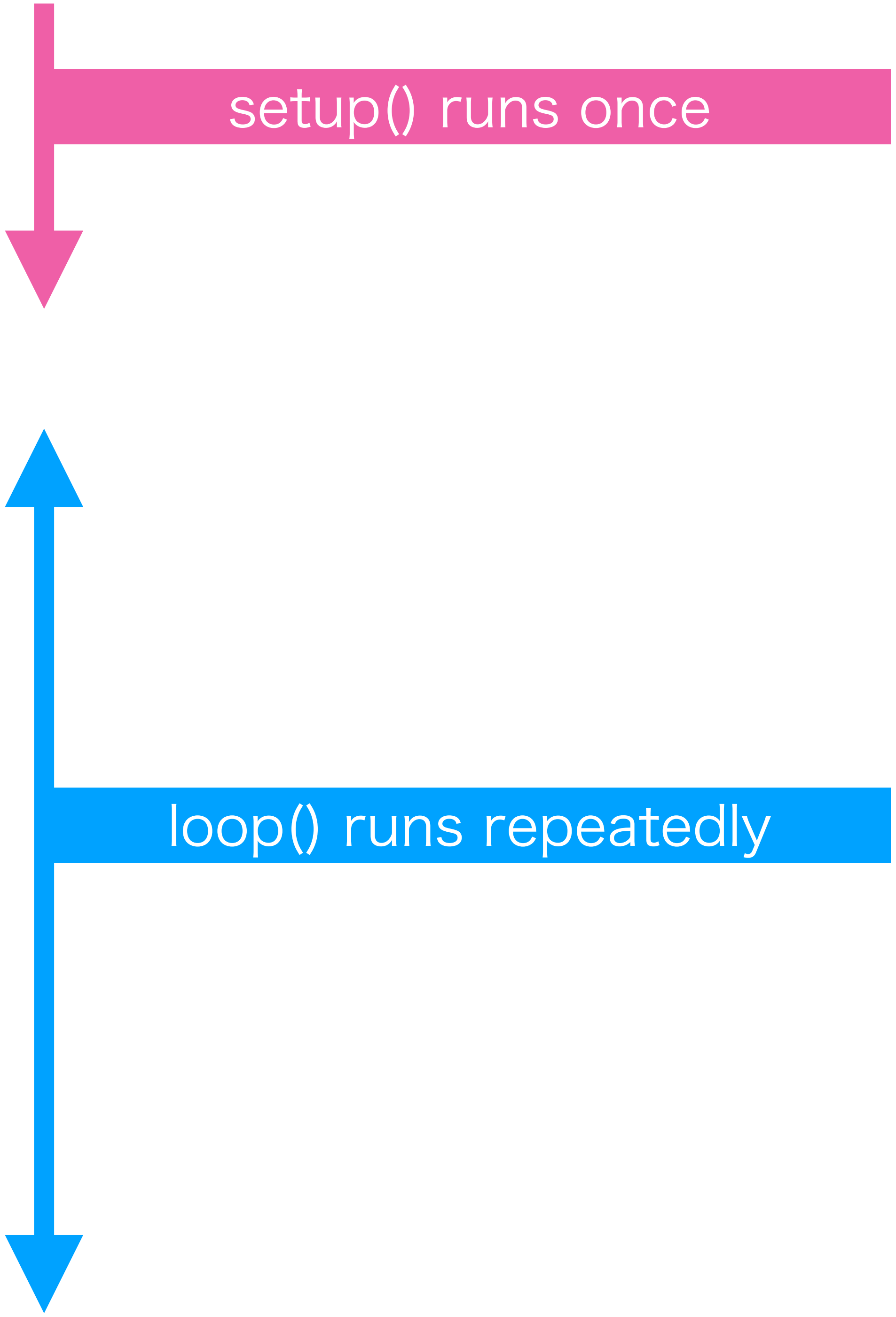
connected
in this direction

Use 13, 12, 11 pins

Try 3-LED Blinker

```
void setup()
{
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
}

void loop()
{
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
  digitalWrite(12, HIGH);
  delay(1000);
  digitalWrite(12, LOW);
  delay(1000);
  digitalWrite(11, HIGH);
  delay(1000);
  digitalWrite(11, LOW);
  delay(1000);
}
```



setup() runs once

loop() runs repeatedly

Conclusions

- Hardware architecture
  - Relationship between a processor and peripheral devices
  - GPIO as an input/output device

- **Next week**:
  Architecture of embedded systems (continued)
- **Homework**:
  Try a 3-LED blinker and then try a 6-LED blinker with different light patterns.