北京大学
PEKING UNIVERSITY

10010010

嵌入式系统编程与实践

2-从信息物理系统到嵌入式系统

燕博南

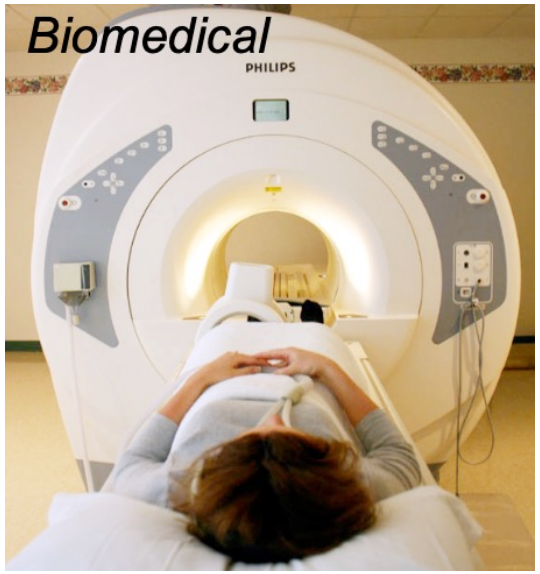
2025秋

Outline

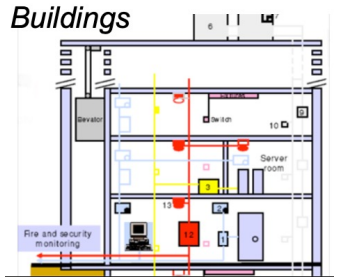
- Introduction to Cyber-Physical System (CPS)
- Introduction to Embedded System
- Embedded System Hierarchy
- Development Flow of Embedded System

Cyber-Physical System (CPS)

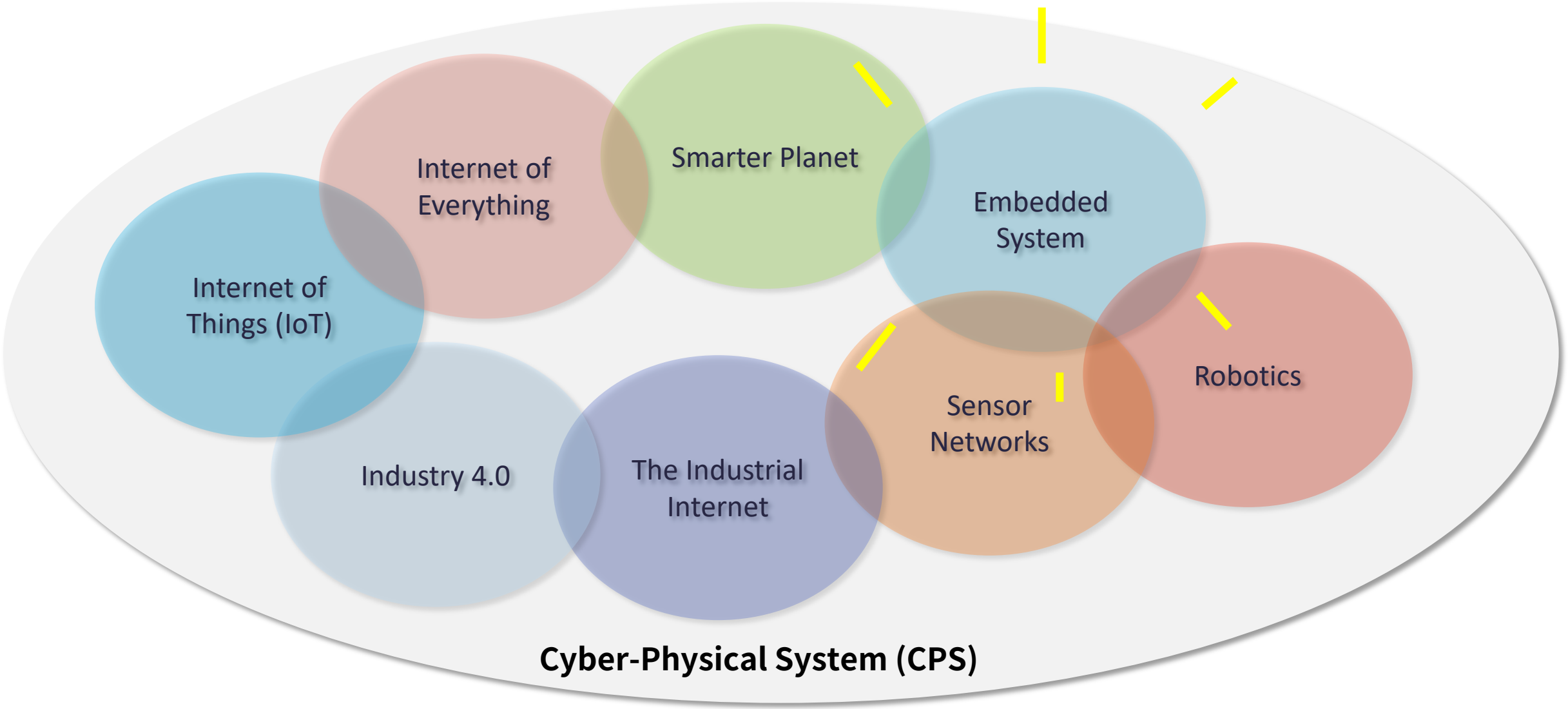
- Definition:
- Integration of computation with physical processes, defined by the intersection



Manufacturing

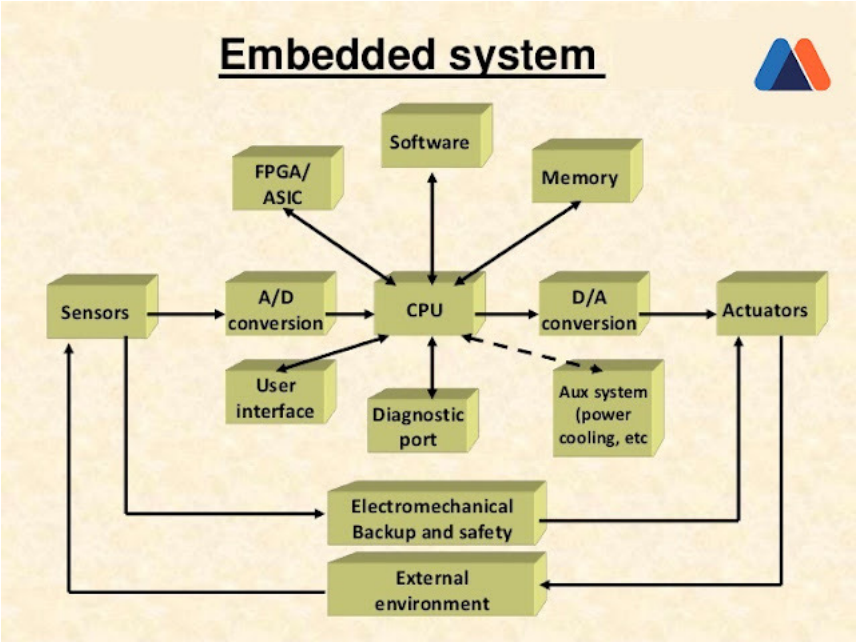


Cyber-Physical System (CPS)



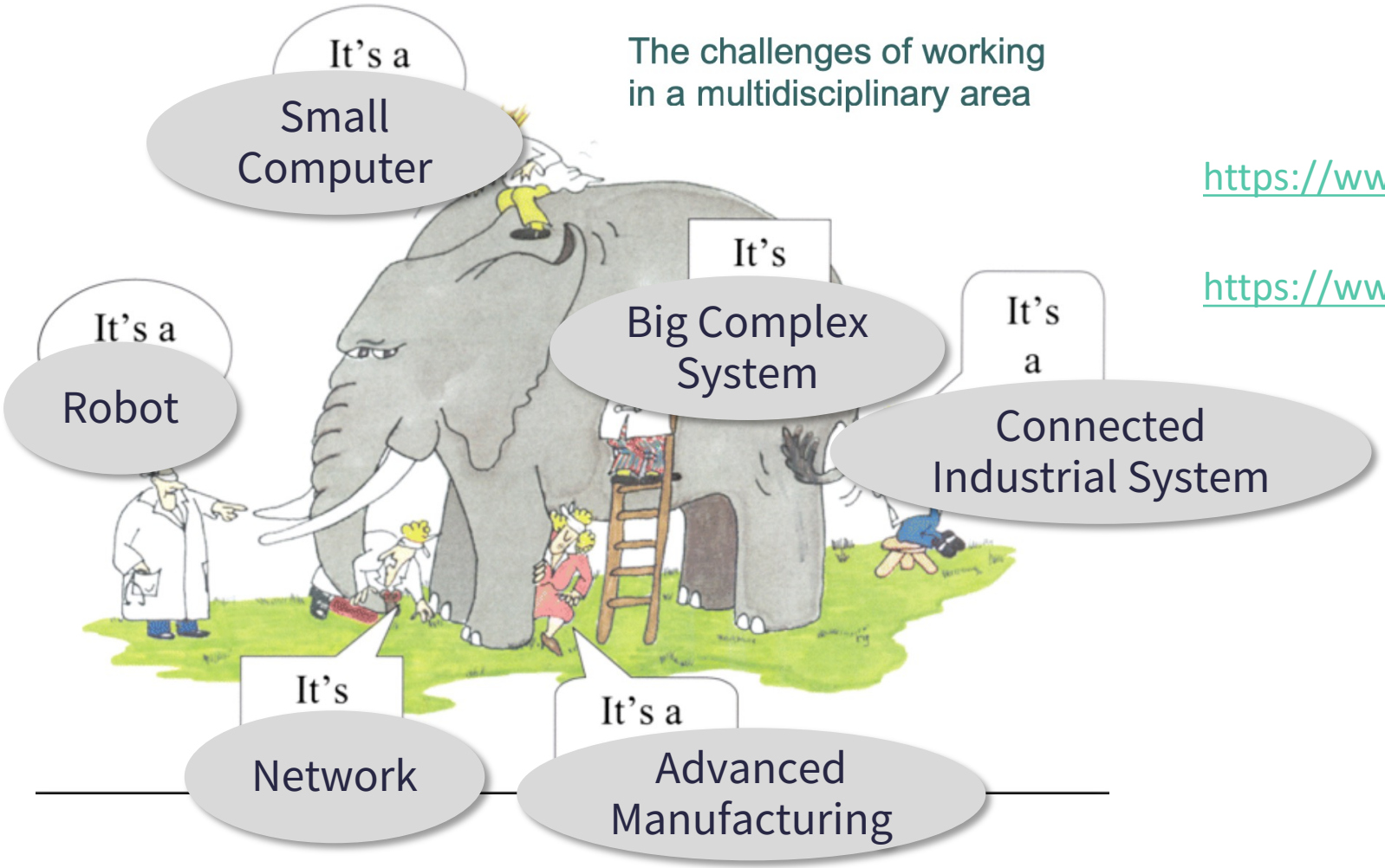
Embedded System

- An **embedded system** is a computer system—a combination of a computer processor, computer memory, and input/output peripheral devices—that has a dedicated function within a larger mechanical or electronic system.



Source: https://www.heavy.ai/technical-glossary/embedded-systems

CPS is A Multidisciplinary Area

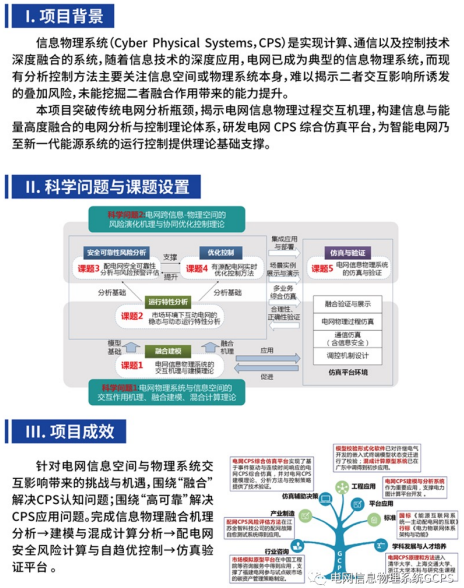


https://www.bilibili.com/video/BV1ZA411e7Ff/

https://www.bilibili.com/video/BV1N14y1E741/

Cyber-Physical System (CPS)

- DARPA: Symbiotic Design for Cyber Physical Systems (SDCPS), $8.7M
- NSF: The CPS program aims to develop the core research needed to engineer these complex CPS, some of which may also require dependable, high-confidence, or provable behaviors , $32.4M
- 我国重点研发计划：电网信息物理系统分析与控制的基础理论与方法



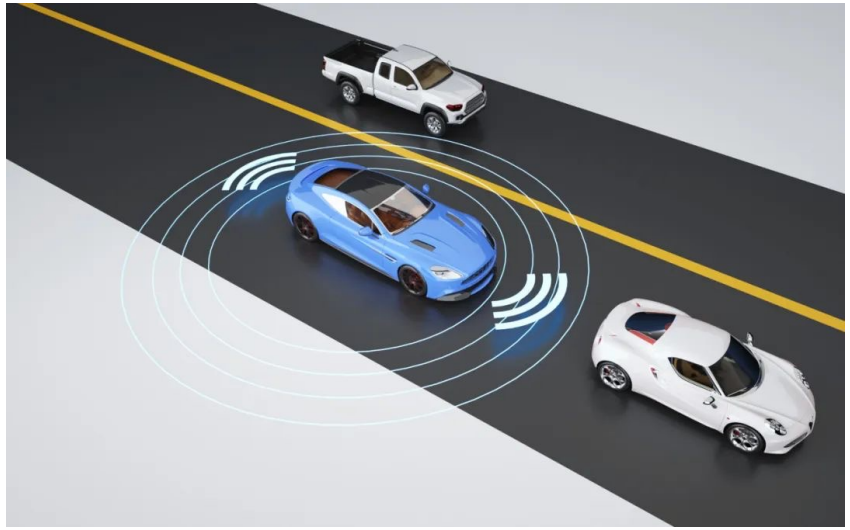
中华人民共和国科学技术部
Ministry of Science and Technology of the People's Republic of China

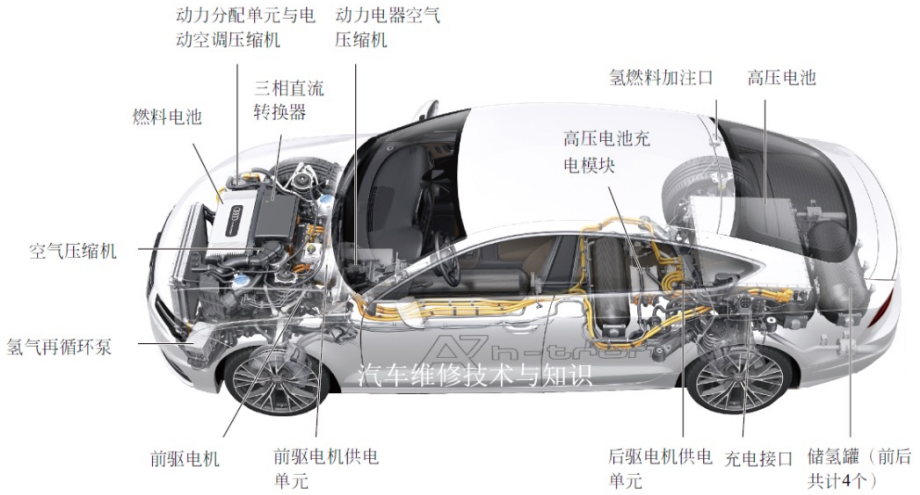
DARPA
DEFENSE ADVANCED RESEARCH PROJECTS AGENCY

NSF National Science Foundation

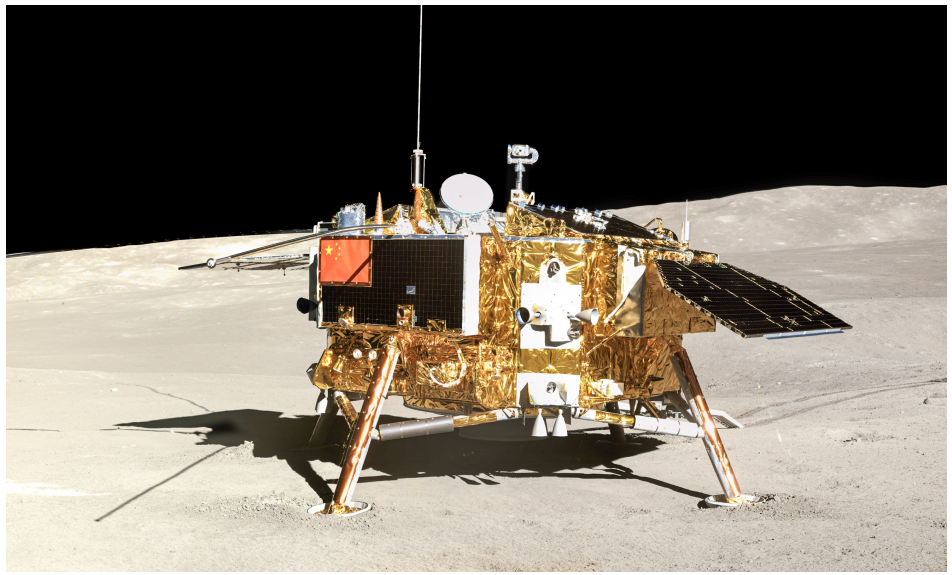
Example: Automotive CPS

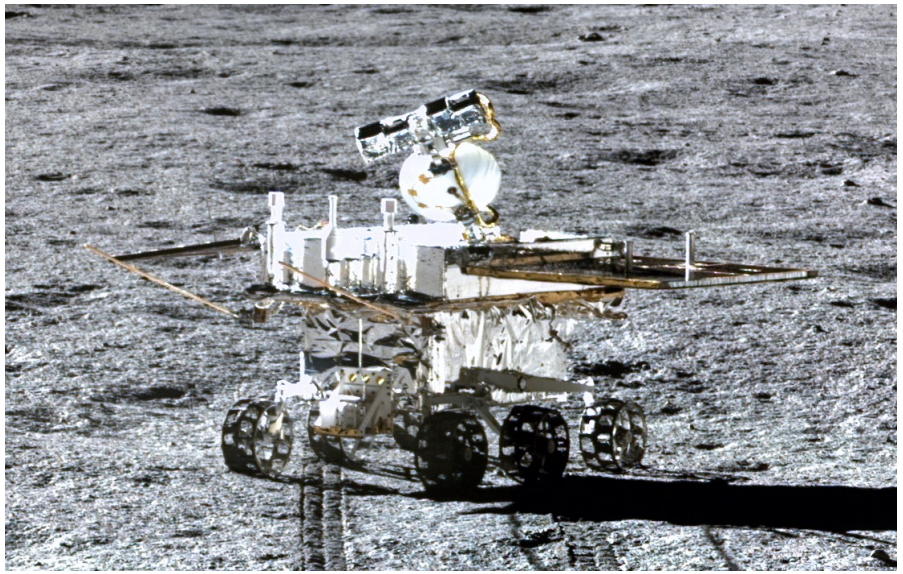
- Safer Transportation
- Reduced Emissions
- Smart Transportation
- Energy Efficiency
- Climate Change
- Human-Robot Collaboration

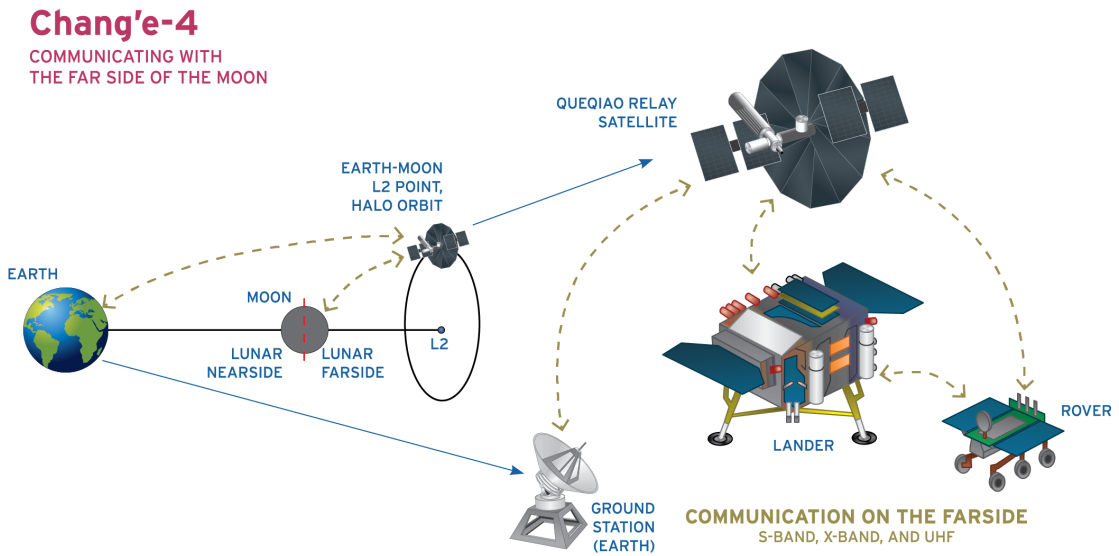




Example: 嫦娥四号着陆器+探测器

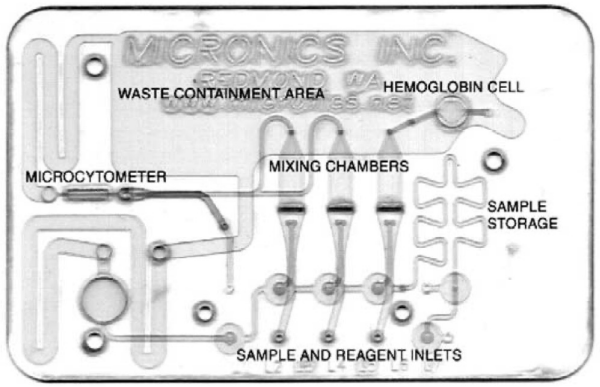


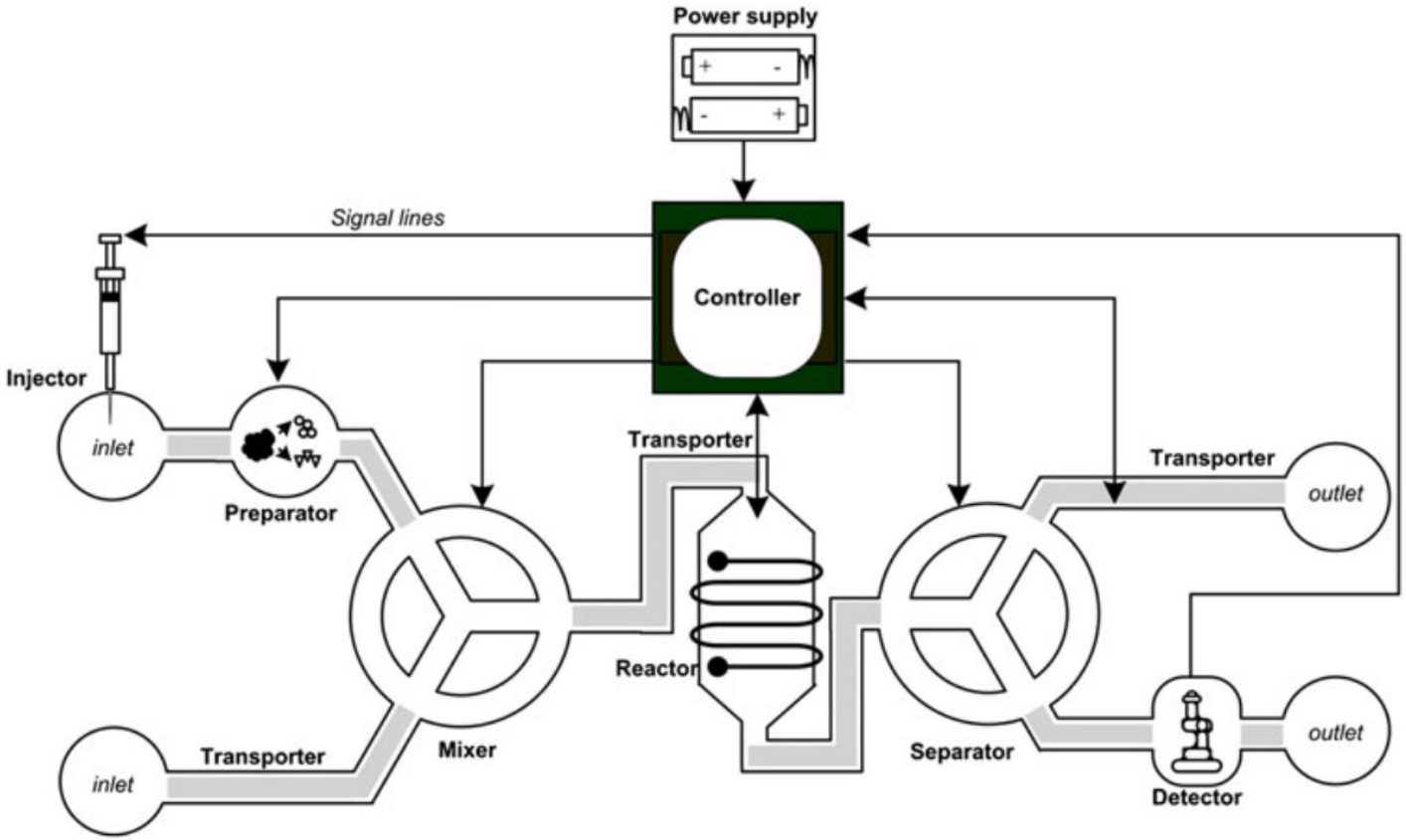




Example: Lab-on-a-chip

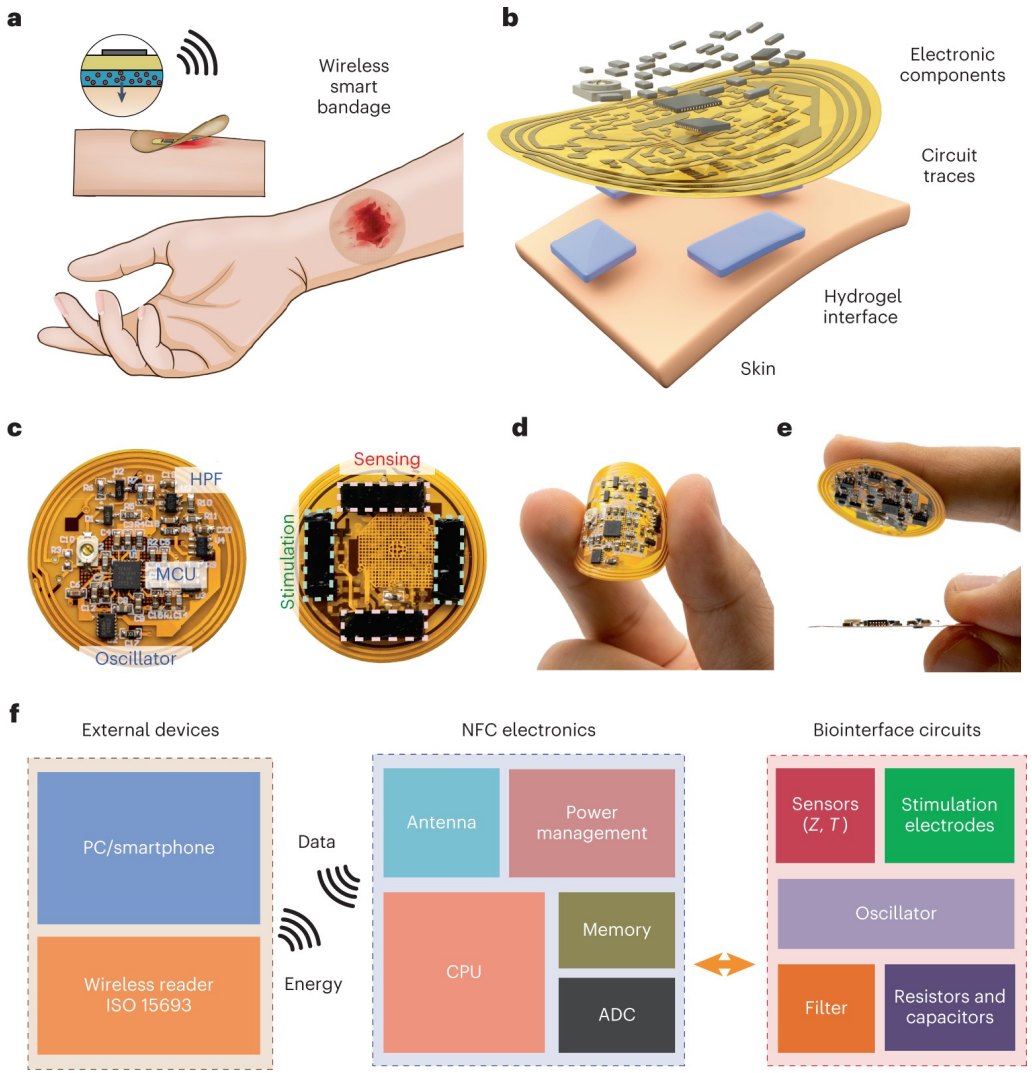
- Precise droplet control
- Real-time analysis





Source: Lab-on-a-chip: a component view

Example: Smart Bandage



- Intersection of :
- Material science
- Electronics
- Embedded Systems

Jiang, Y., Trotsyuk, A.A., Niu, S. *et al.Nat Biotechnol* **41**, 652–662 (2023).

Development Flow of CPS

- **Modeling** is the process of gaining a deeper understanding of a system through imitation. Models express **what** a system does or should do.

- **Design** is the structured creation of artifacts. It specifies **how** a system does what it does.

- **Analysis** is the process of gaining a deeper understanding of a system through dissection. It specifies **why** a system does what it does (or fails to do what a model says it should do).

Motivating Example of a Cyber-Physical System

- (see Chapter 1 in book)

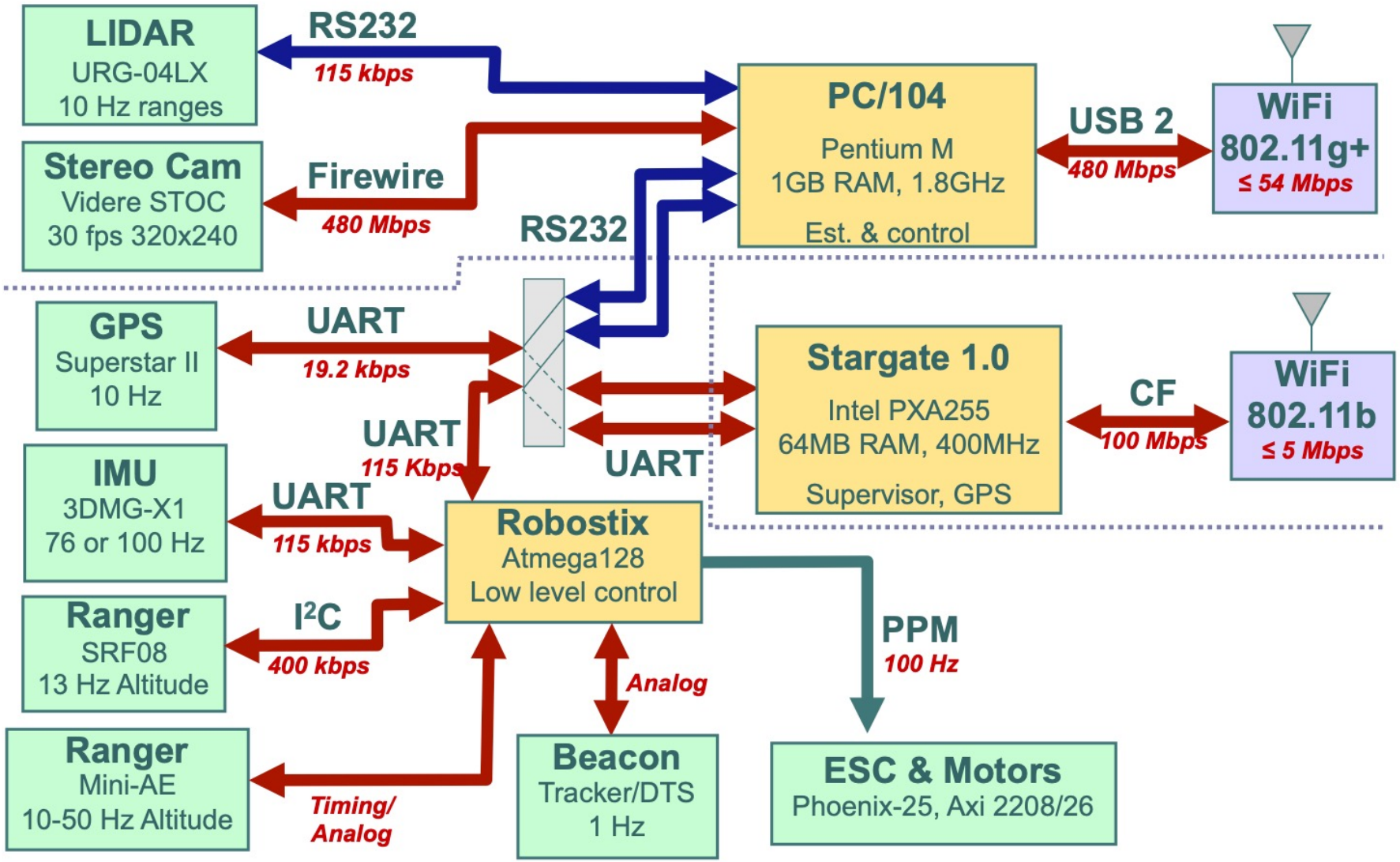
STARMAC quadrotor aircraft (Tomlin, et al.)
多旋翼无人机

Modeling:
- Flight dynamics (ch2)
- Modes of operation (ch3)
- Transitions between modes (ch4)
- Composition of behaviors (ch5)
- Multi-vehicle interaction (ch6)

Design:
- Sensors and Actuators (ch7)
- Processors (ch8)
- Memory system (ch9)
- Sensor interfacing (ch10)
- Concurrent software (ch11)
- Real-time scheduling (ch12)

Analysis
- Specifying safe behavior (ch13)
- Achieving safe behavior (ch14)
- Verifying safe behavior (ch15)
- Guaranteeing timeliness (ch16)
- Security and privacy (ch17)

Design of STARMAC Design



Think Critically

- Any course that purports to teach you how to design embedded systems is misleading you
- The technology will change, rapidly!
- Our goal is to teach you how things are done today, and why that is not good enough.

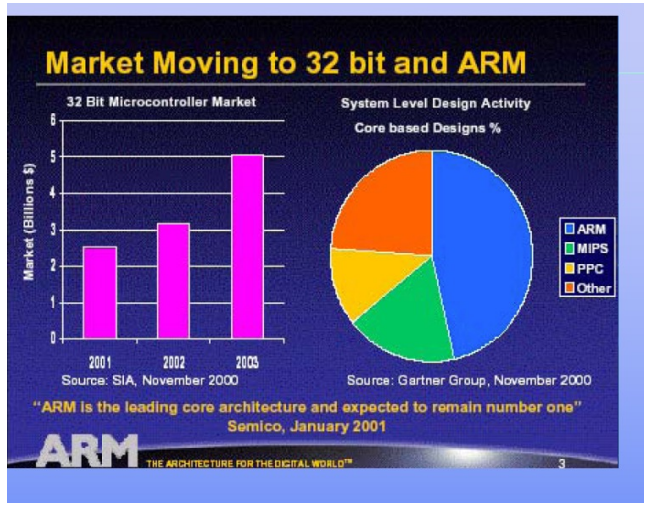
Courtesy: UC Berkeley EECS149

Outline

- Introduction to Cyber-Physical System (CPS)
- Introduction to Embedded System
- Embedded System Hierarchy
- Development Flow of Embedded System

Historic Perspective: Embedded System Depends on IC Technology

嵌入式系统教学讲义（何广强）



计算机的发展

Apple Macintosh
黑白显示器
1 MIP 处理器

Acorn Archimedes
彩色显示器
10 MIP 处理器
... RISC集

惠普一康柏掌上电脑
彩色显示
200 MIP 处理器
64MB 内存
...一袖珍型

1.3.5嵌入式外围接口电路和设备接口

根据外围设备的功能可分为以下5类

- 存储器类型
- 通信接口
- 输入输出设备
- 设备扩展接口
- 电源及辅助设备

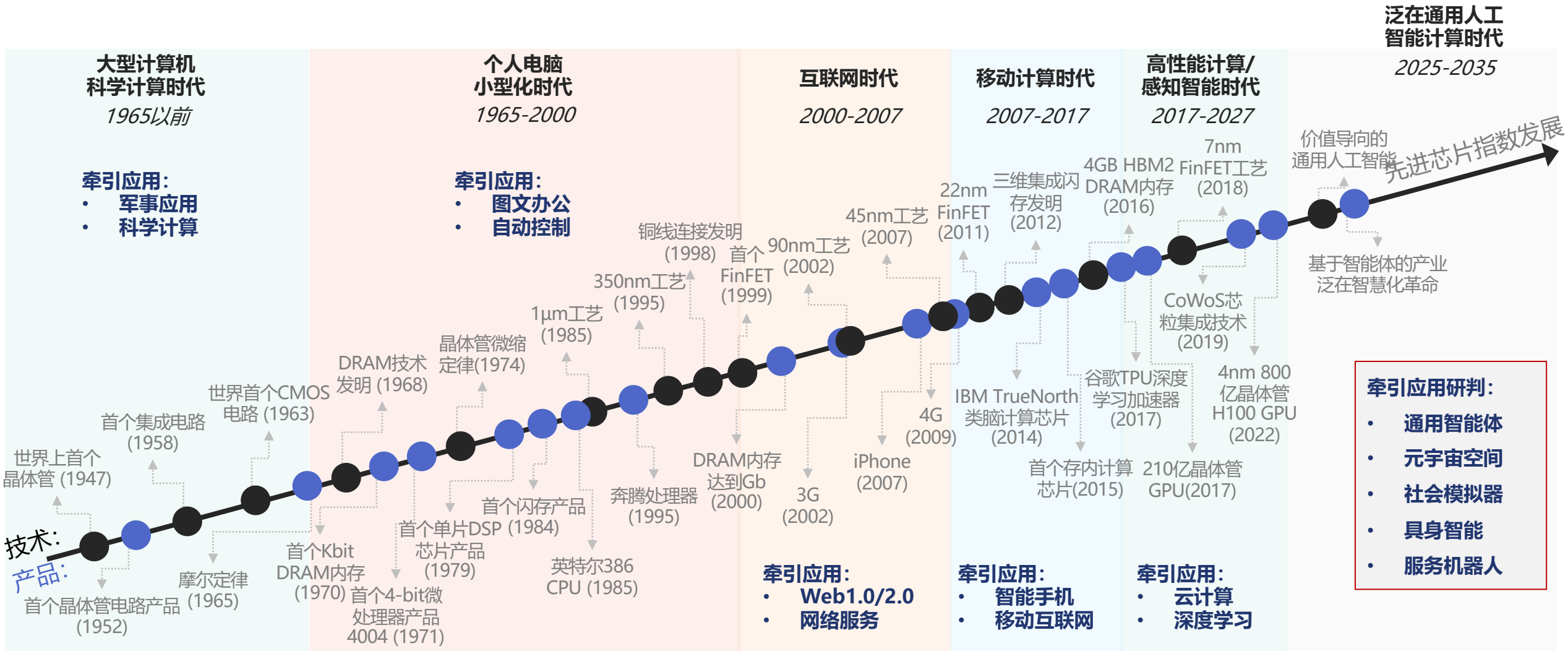
1.3.1嵌入式系统软件的层次结构

具有操作系统的嵌入式软件层次

- 驱动层程序
- 实时操作系统（RTOS）
- 操作系统的应用程序接口（API）
- 应用程序

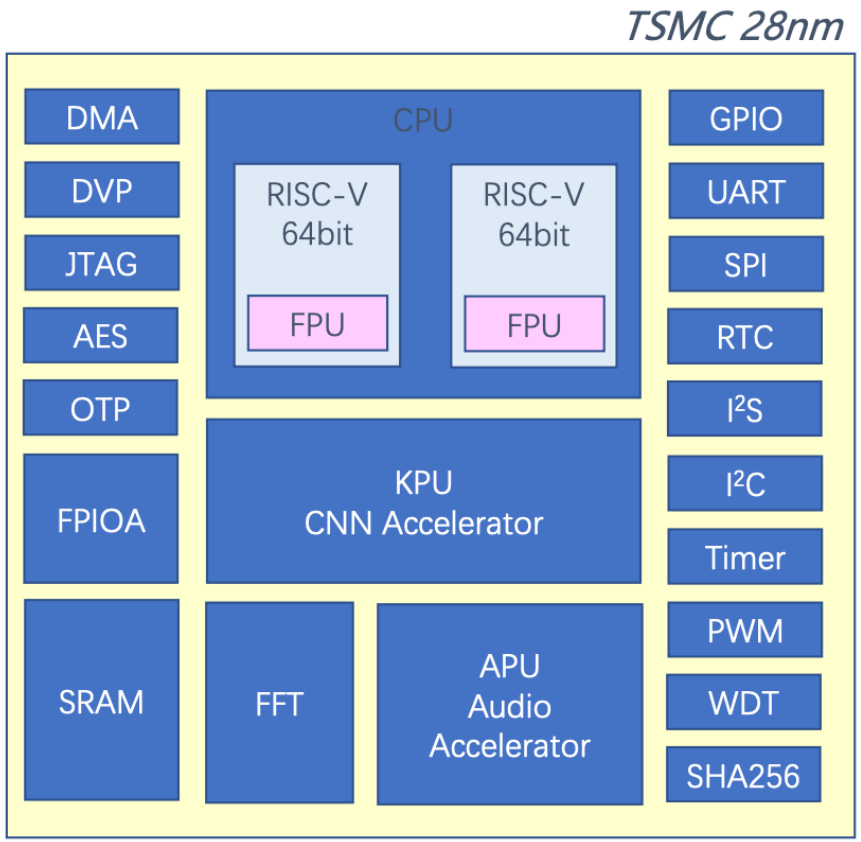
1.3.2 启动程序BootLoader介绍

Historic Perspective: Embedded System Depends on IC Technology



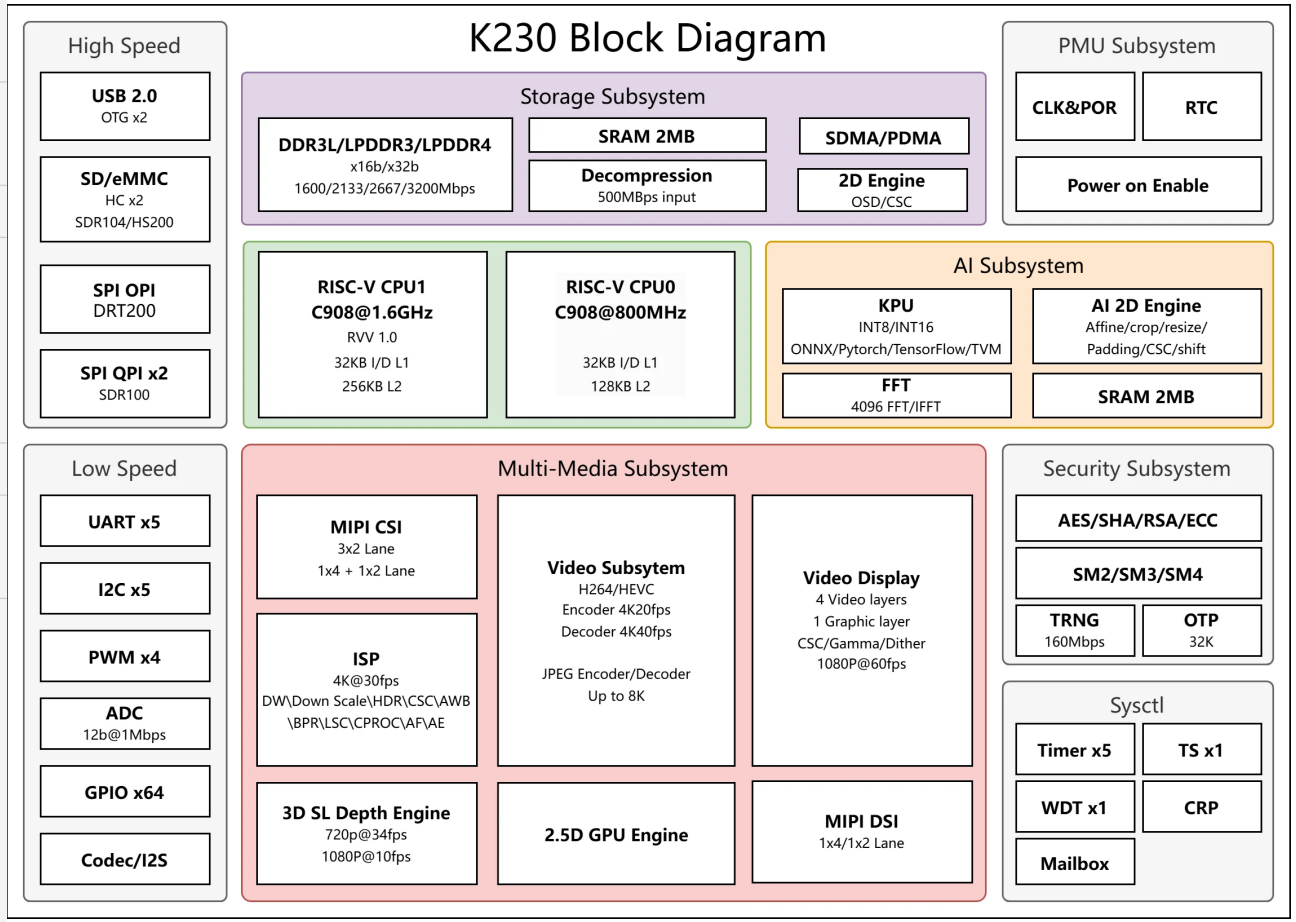
K210 Architecture

- CPU x2
  - 额定运行频率400MHz，各自带独立FPU（单精度浮点）
- KPU x1：神经网络加速器
  - 卷积、池化、激活
- APU x1：语音处理单元
  - Beam-forming、降采样、FIR、FFT
  - 最多支持8路音频数据，16个方向
- SRAM x1：8MB
  - 其中2MB为KPU专用，剩余6MB由用户程序和模型数据共用
- DVP x1：摄像头接口（8位并口）
  - 输入：RGB565、YUV422
  - 输出：RGB565、RGB888、YUV422的Y分量
  - 支持的最大分辨率1920*1023
- DMA x6：6个通道



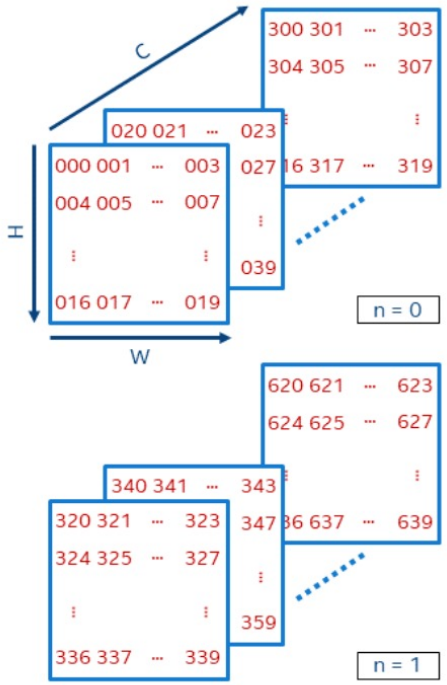
K230 Architecture

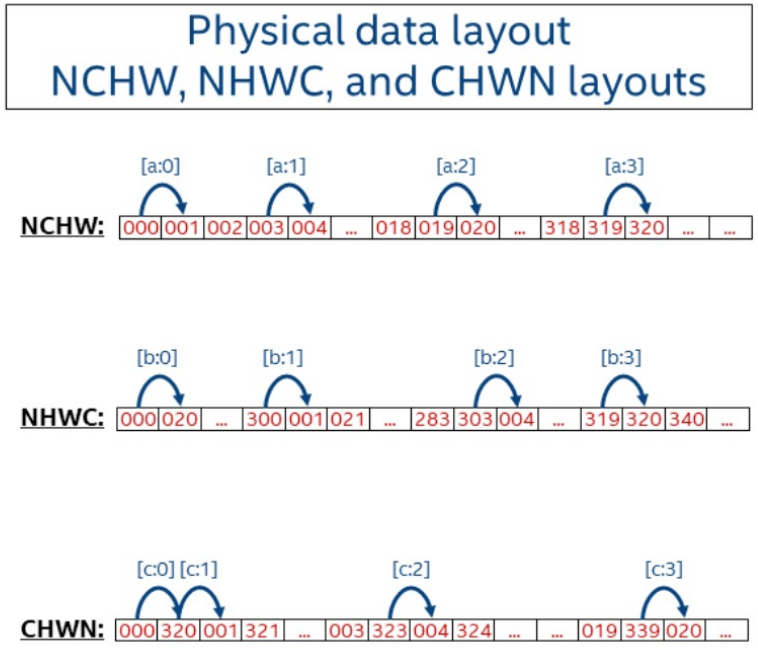
设备参数	
CPU	CPU 1: RISC-V 处理器，1.6GHz，32KB I-cache, 32KB D-cache, 256KB L2 Cache，128bit RVV 1.0扩展
	CPU 0: RISC-V 处理器，0.8GHz，32KB I-cache, 32KB D-cache, 128KB L2 Cache
KPU	支持 INT8 和 INT16
	典型网络性能：Resnet 50 ⩾ 85fps @INT8 ； Mobilenet_v2 ⩾ 670fps @INT8；YoloV5S ⩾ 38fps @INT8
DPU	3D 结构光深度引擎，最大分辨率支持1920*1080
VPU	H.264 和 H.265 编码器和解码器最大分辨率支持4096*4096
	编码器性能：3840*2160@20fps
	解码器性能：3840*2160@40fps
	JPEG 编解码器：支持最大 8K（8192*8192）分辨率
图像输入	支持最大3路 MIPI CSI(1x4lane+1x2lane) 或 3x2lane
显示输出	1路 MIPI DSI, 1x4 lane 或 1x2 lane
	最大分辨率1920*1080@60fps
片上接口	5 x UART
	5 x I2C
	6 x PWM
	64 x GPIO + 8 x PMU GPIO
	2 x USB 2.0 OTG
	2 x SDxC: SD3.01, eMMC 5.0
	3 x SPI: 1 x OSPI + 2 x QSPI
	WDT / RTC / Timer



KPU运行卷积深度学习网络

- 网络输入或者中间的feature map尺寸不大于320*256，不小于4*4
- 与tensorflow支持的数据排布格式NHWC不同，KPU支持的数据排布格式为NCHW





KPU

- 支持的普通二维卷积

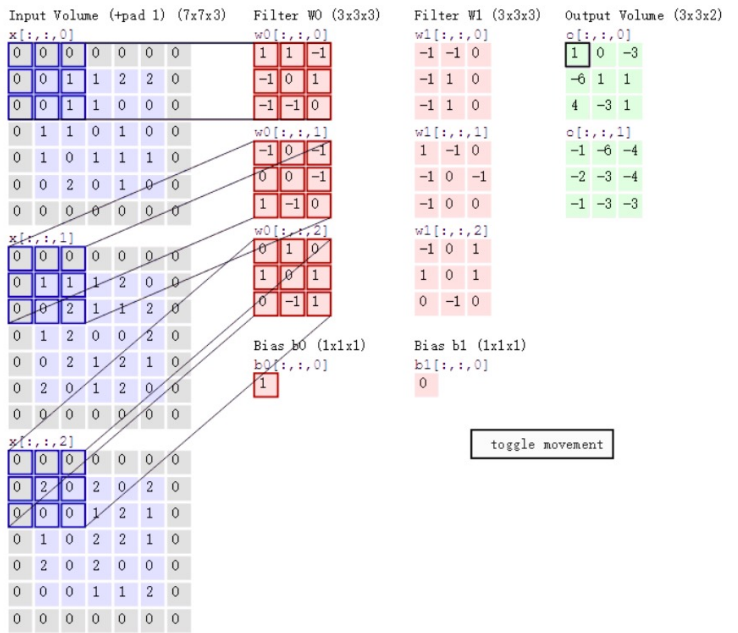
ksize	stride	required tensorflow padding
1*1	1	same或valid
3*3	1	same
3*3	2	卷积前将feature map手动padding一圈，然后卷积时padding方式用valid

- 支持的depthwise二维卷积

ksize	stride	required tensorflow padding
3*3	1	same
3*3	2	卷积前将feature map手动padding一圈，然后卷积时padding方式用valid

- 支持的pooling

ksize	stride	required tensorflow padding
2*2	1	same
2*2	2	如果feature map size是ksize的整数倍，same或valid；否则valid
4*4	4	如果feature map size是ksize的整数倍，same或valid；否则valid



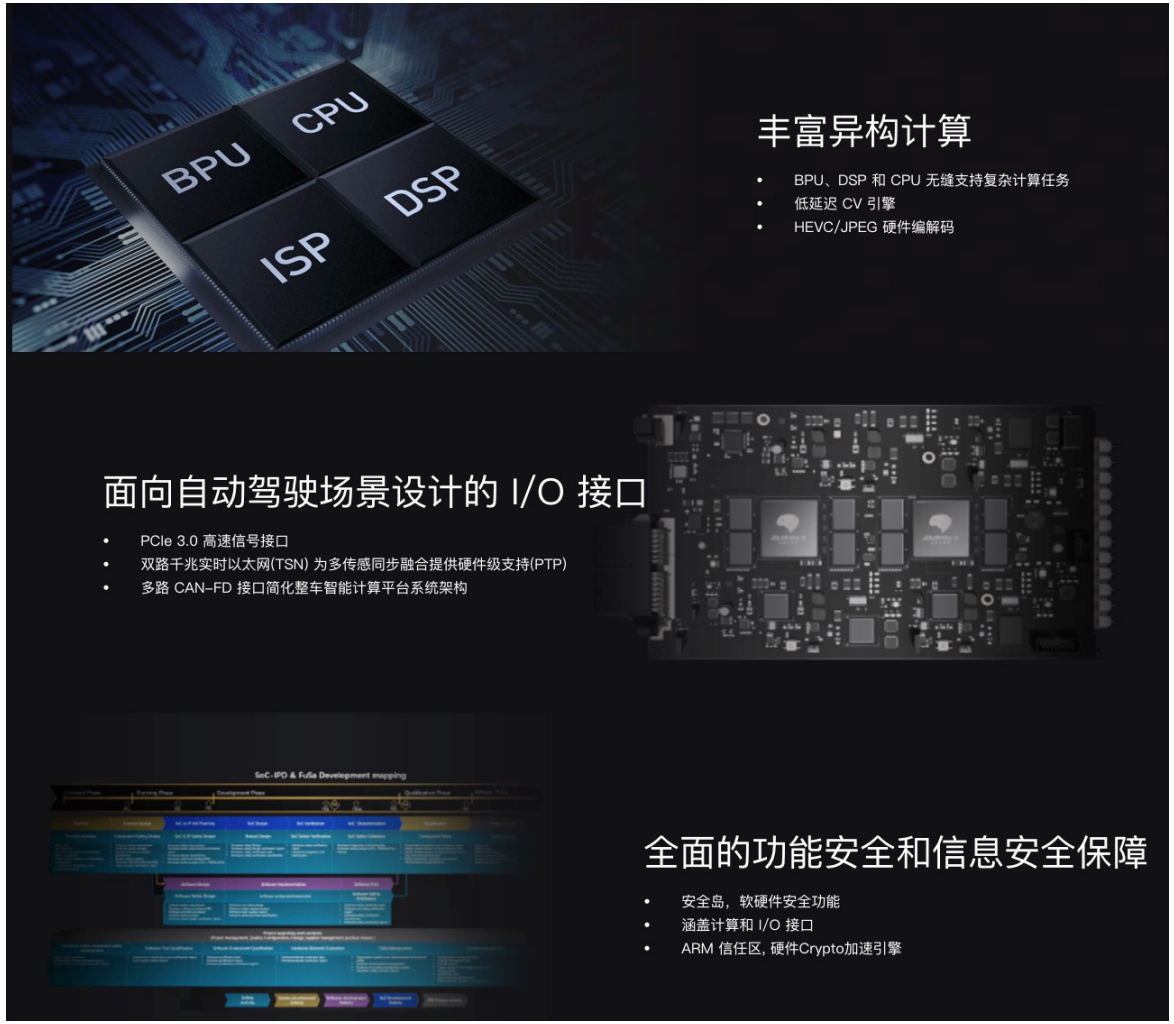
KPU工作流程

用于将主流训练框架按照特定限制规则训练出来的浮点模型量化转化为K210可运行的格式

- 支持的训练框架
  - Tensorflow、Caffe、Paddlepaddle
- 量化位数
  - 8bit、16bit
- 支持的操作
  - Conv2d、DepthwiseConv2d、FullyConnected、Add、MaxPool2d、AveragePool2d、GlobalAveragePool2d、BatchNormalization、BiasAdd、Relu、Relu6、LeakyRelu、Concatenation、L2Normalization、Softmax、Flatten、ResizeNearestNeighbor

其它例子：地平线征程5

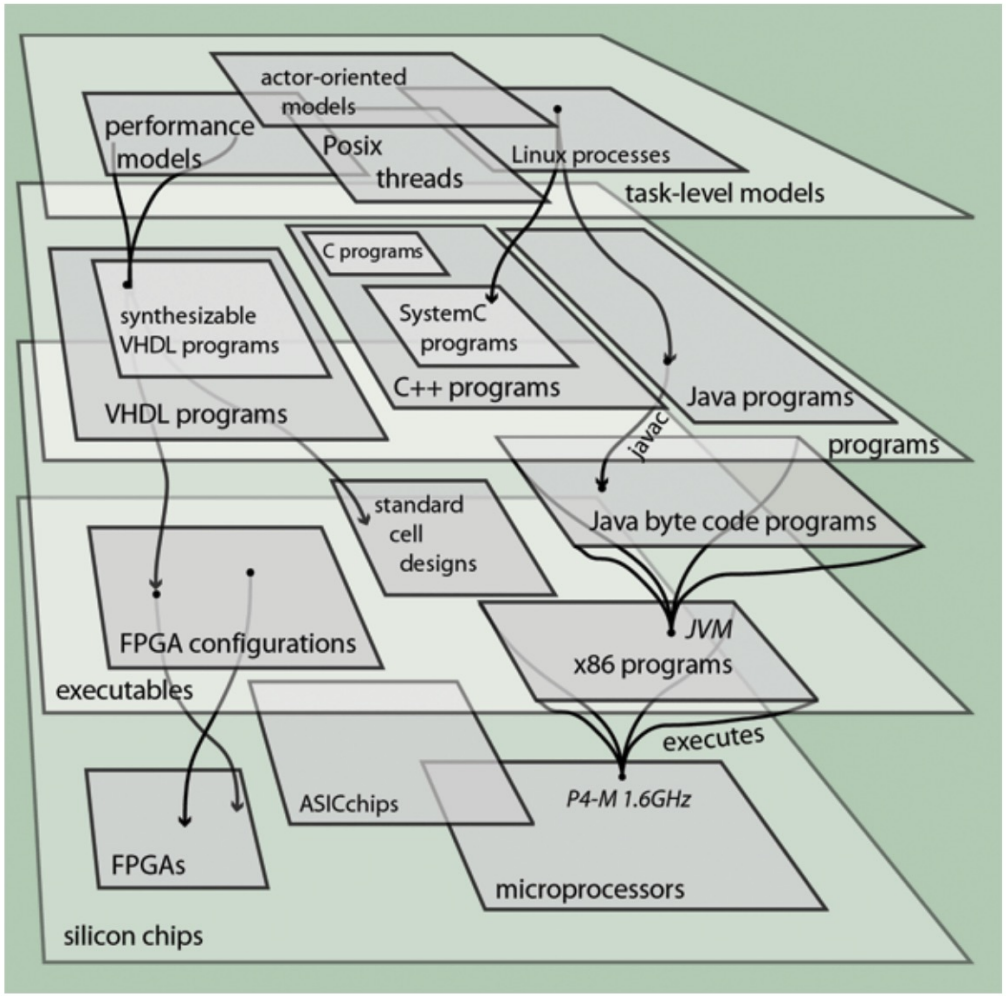

征程5 芯片架构
Video input
MIPI CSI-2 RX Parallel
Video output
MIPI CSI-2 TX
Engine
CV Engine
ISP
Dual-core ISP
DSP
Dual-core Vision DSP
CPU
Octa-core CPU
Deep Learning engine
Dual-core BPU Deep Learning engine
Codec
HEVC /MJPEG Codec
Advanced Security
Crypto engine
Secure OTA
Secure boot
Safety island
Dual-core lockstep MCU
Memory sub-system
DDR4/LP4/LP4x
SD/SDIO/eMMC
NOR/SPI
NAND/Hyperflash
Connectivity
GbE (TSN)
PCIe
LPWM & PWM
CAN FD
SPI
UART
I2C
I2S
16nm FinFET
27×27mm HFCBGA
AEC-Q100 Grade 2
ASIL-B


BPU
CPU
ISP
DSP
丰富异构计算
BPU、DSP 和 CPU 无缝支持复杂计算任务
低延迟 CV 引擎
HEVC/JPEG 硬件编解码
面向自动驾驶场景设计的 I/O 接口
PCIe 3.0 高速信号接口
双路千兆实时以太网(TSN) 为多传感同步融合提供硬件级支持(PTP)
多路 CAN-FD 接口简化整车智能计算平台系统架构
全面的功能安全和信息安全保障
安全岛，软硬件安全功能
涵盖计算和 I/O 接口
ARM 信任区, 硬件Crypto加速引擎

Outline

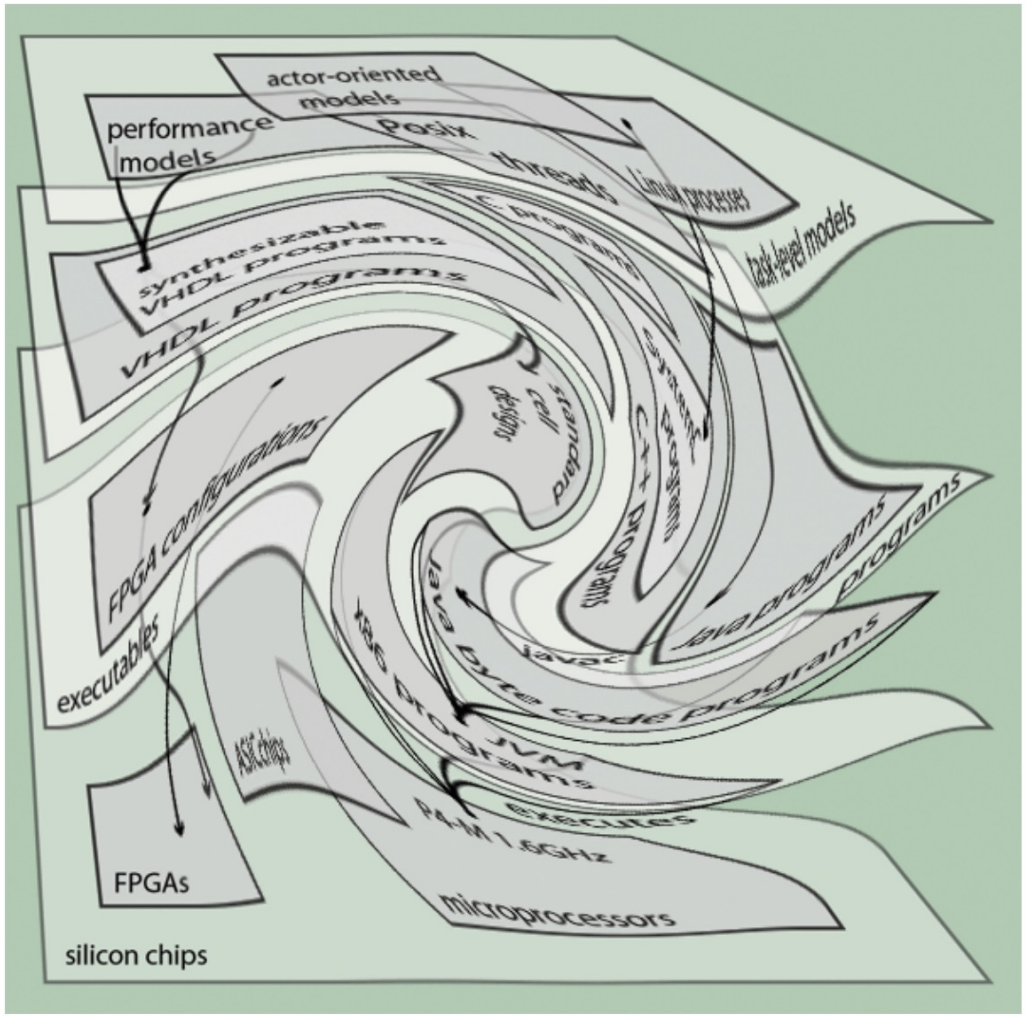
- Introduction to Cyber-Physical System (CPS)
- Introduction to Embedded System
- Embedded System Hierarchy
- Development Flow of Embedded System

Embedded System Abstraction



The purpose of an abstraction is to hide details of the implementation below and provide a platform for design from above.

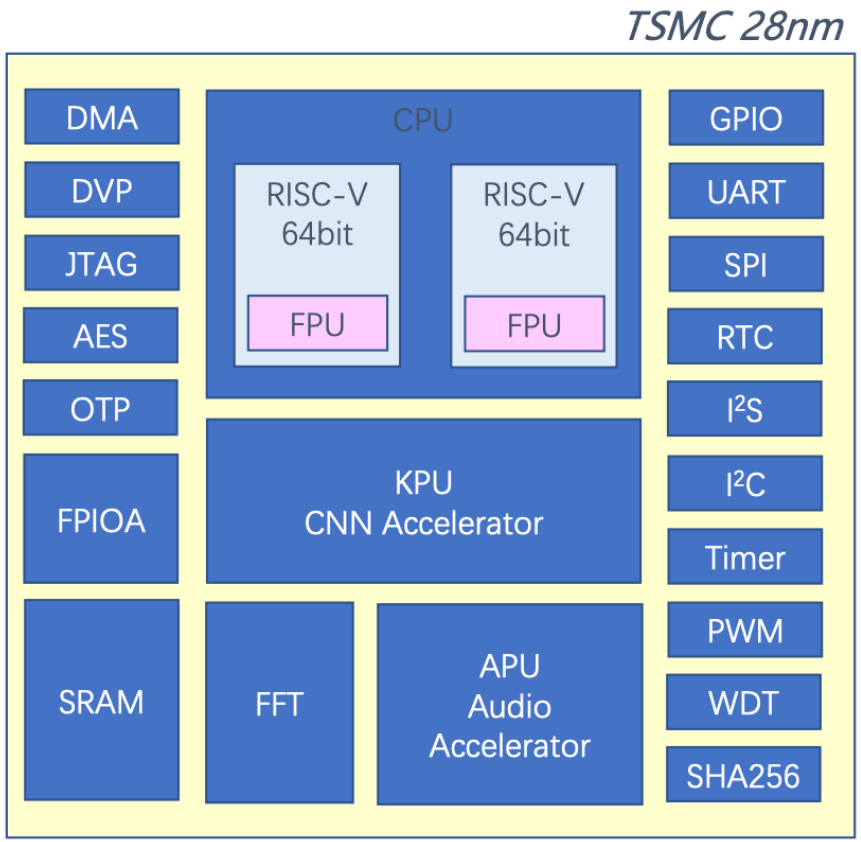
Embedded System Abstraction



Sometimes it is too hard to isolate layers…

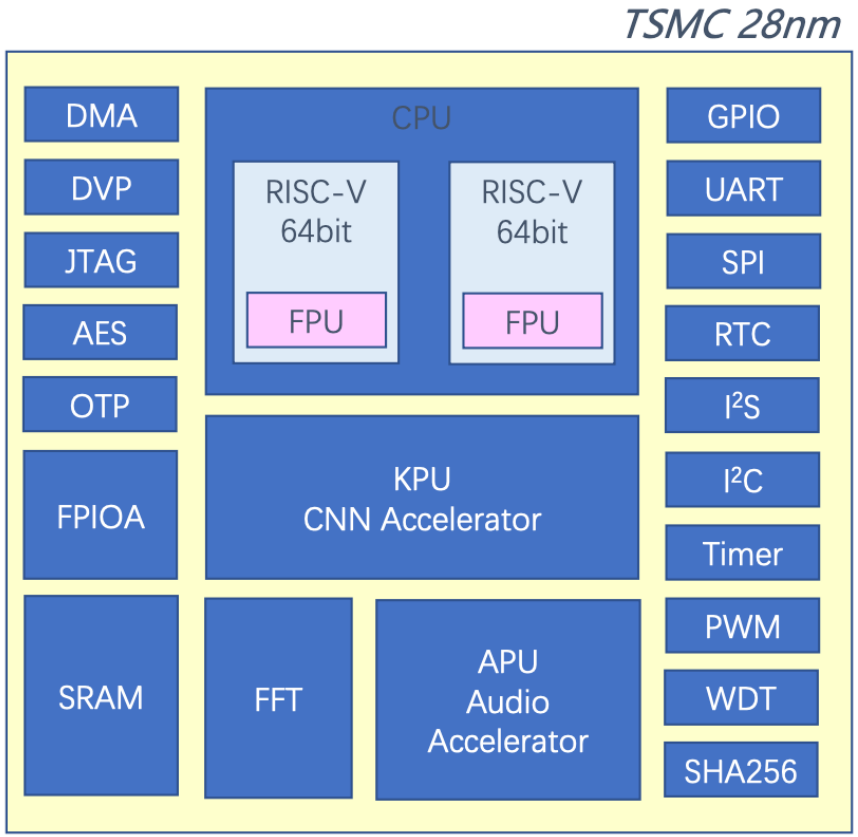
K210 Hardware (Revisit)

- CPU x2
  - 额定运行频率400MHz，各自带独立FPU（单精度浮点）
- KPU x1：神经网络加速器
  - 卷积、池化、激活
- APU x1：语音处理单元
  - Beam-forming、降采样、FIR、FFT
  - 最多支持8路音频数据，16个方向
- SRAM x1：8MB
  - 其中2MB为KPU专用，剩余6MB由用户程序和模型数据共用
- DVP x1：摄像头接口（8位并口）
  - 输入：RGB565、YUV422
  - 输出：RGB565、RGB888、YUV422的Y分量
  - 支持的最大分辨率1920*1023
- DMA x6：6个通道

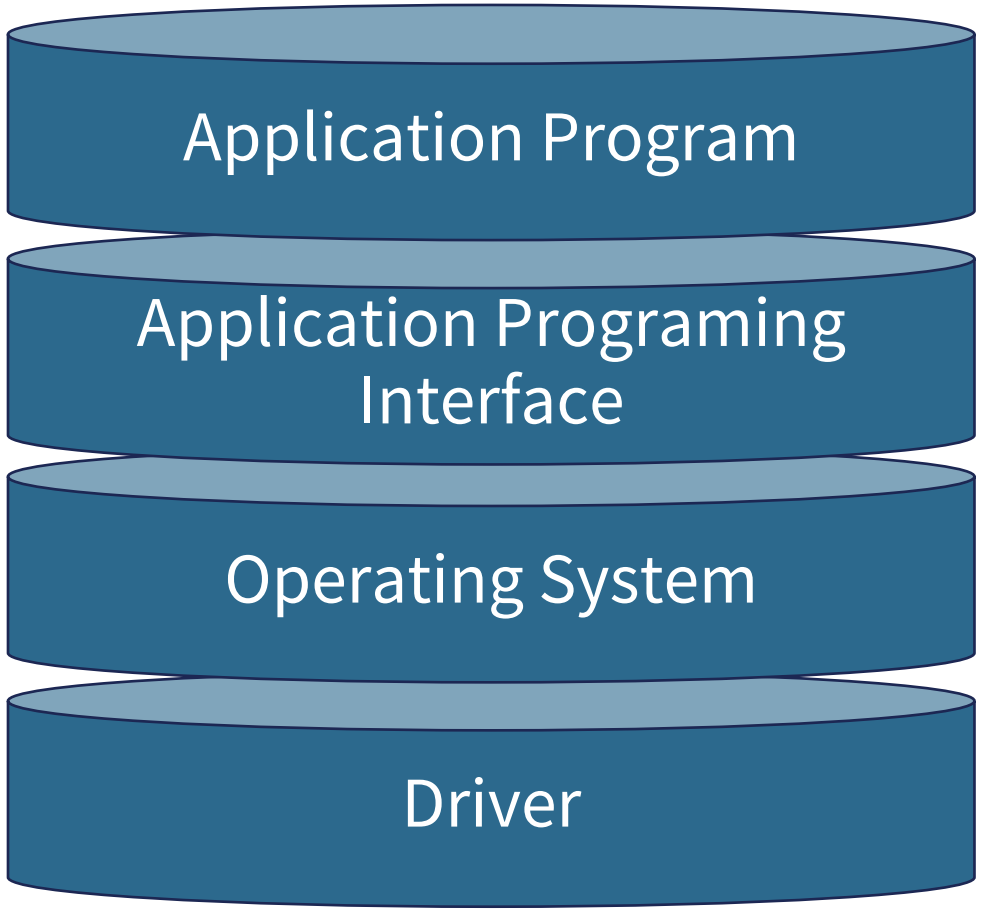


K210 Hardware (Revisit)

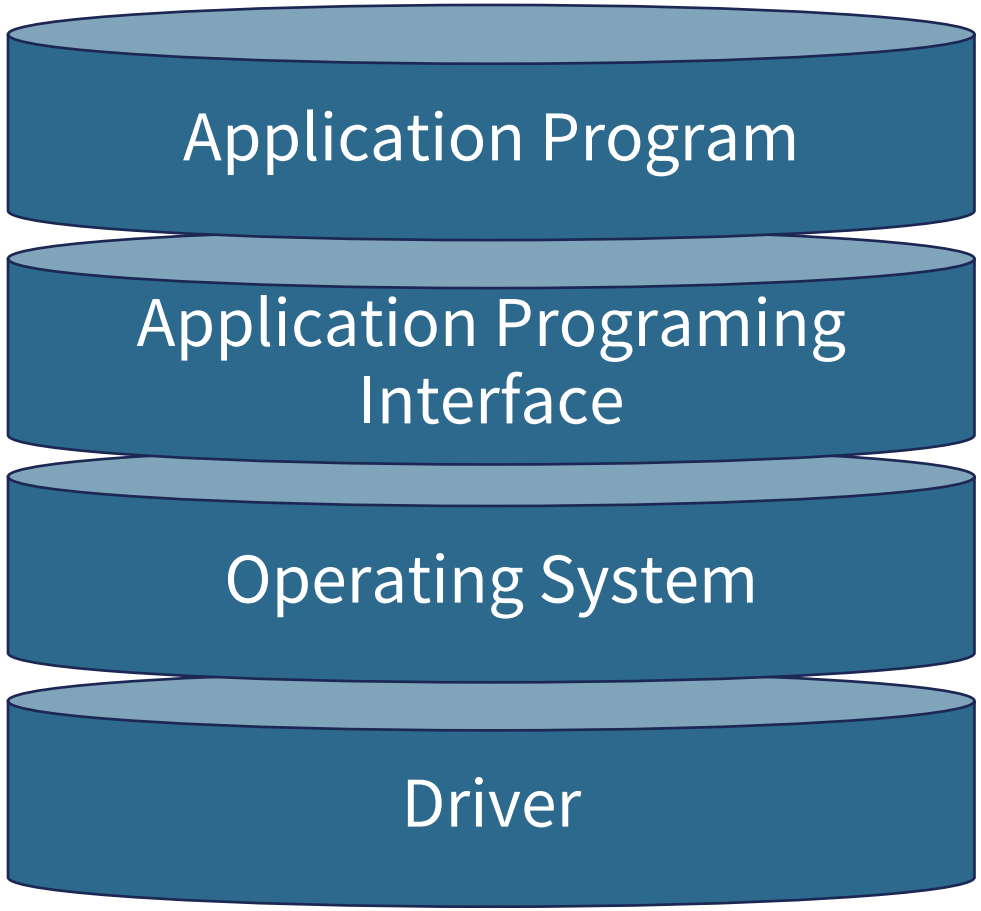
- FPIOA x1：现场可编程IO阵列
  - 用于将芯片内部功能映射到48个物理IO上
- GPIO x40
- UART x4
  - 推荐使用UART1、UART2、UART3，波特率达5Mbps
  - 推荐UART0仅用于调试
- SPI x4
  - SPI0、SPI1、SPI3为master，SPI2为slave
  - SPI3固定接片外FLASH，SPI0通常接屏幕
  - Master时钟速度达80MHz，slave时钟速度达30MHz
- I2S x3
  - 每个I2S最多可接8路音频数据
  - I2S0可以连接到APU
- I2C x3
- 定时器 x3，WDT x2，RTC x1
- FFT x1，SHA256 x1，AES x1



Software for Embedded Systems



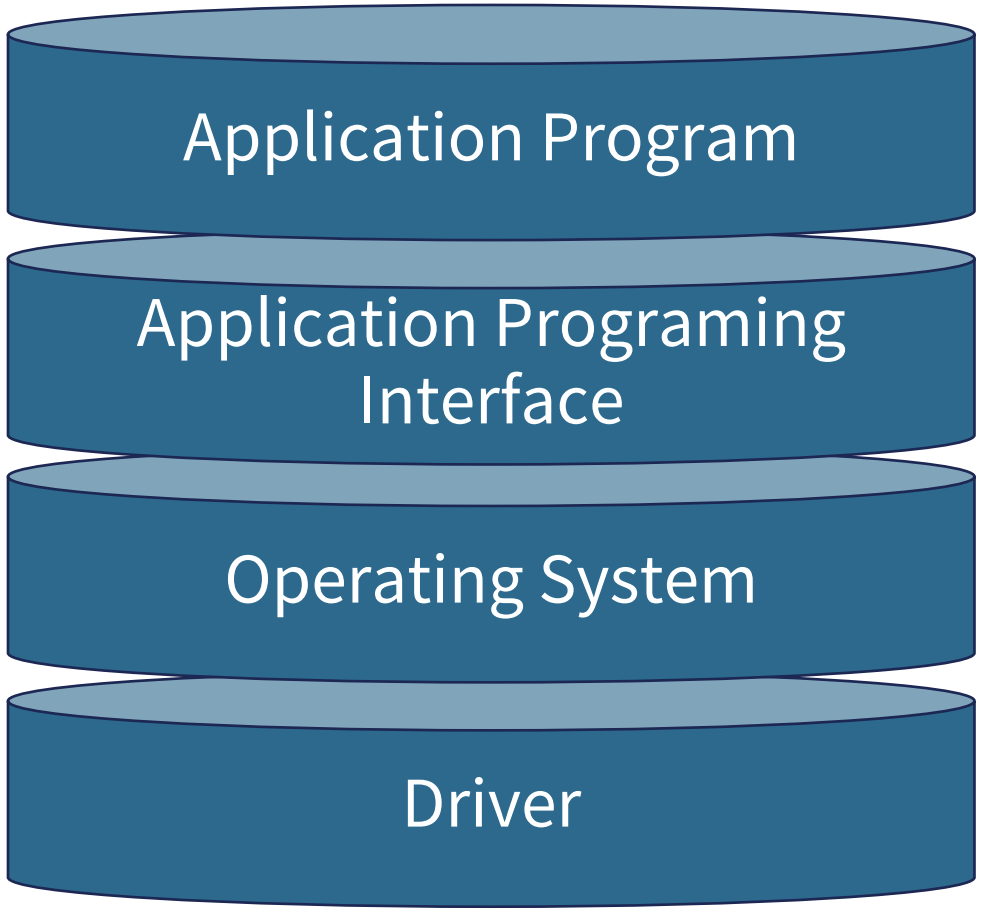
Software for Embedded Systems



Example: FT2232H

- Single chip USB to dual channel UART (RS232, RS422 or RS485).
- Single chip USB to dual channel FIFO.
- Likewise Single chip USB to dual channel JTAG.
- Single chip USB to dual channel SPI.
- Additionally Single chip USB to dual channel I^2C.
- Single chip USB to dual channel Bit-Bang.

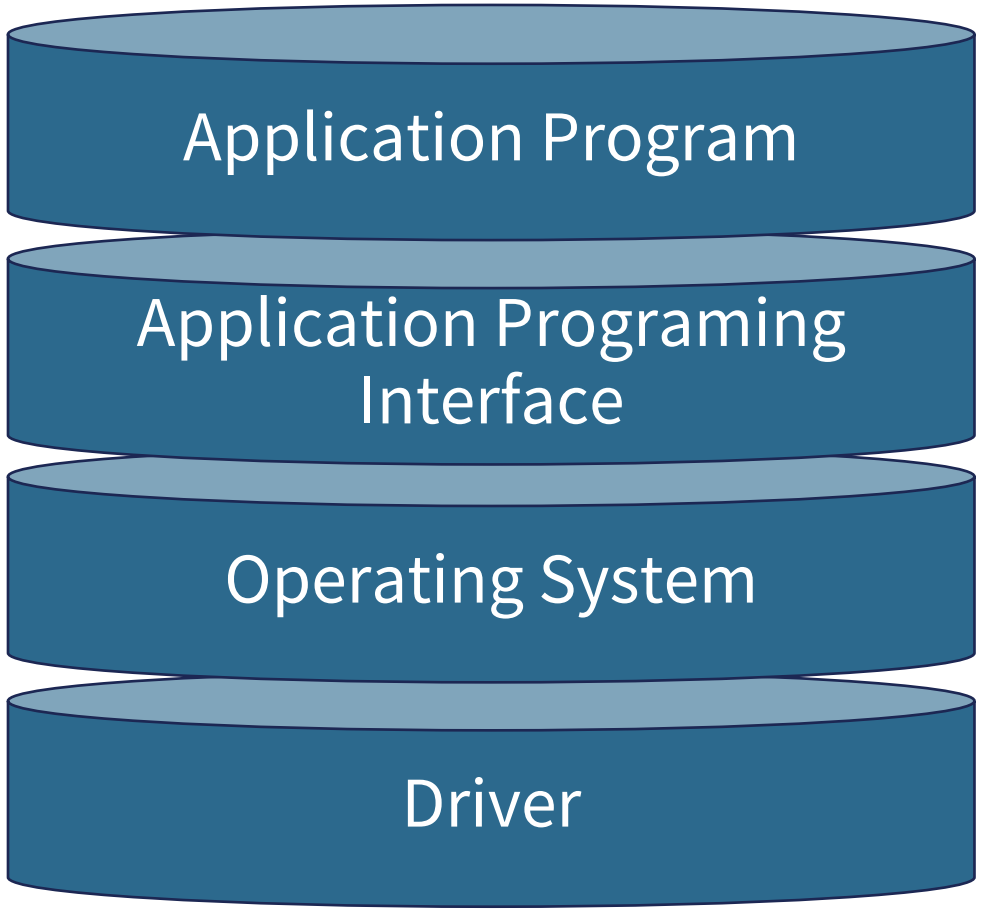
Software for Embedded Systems

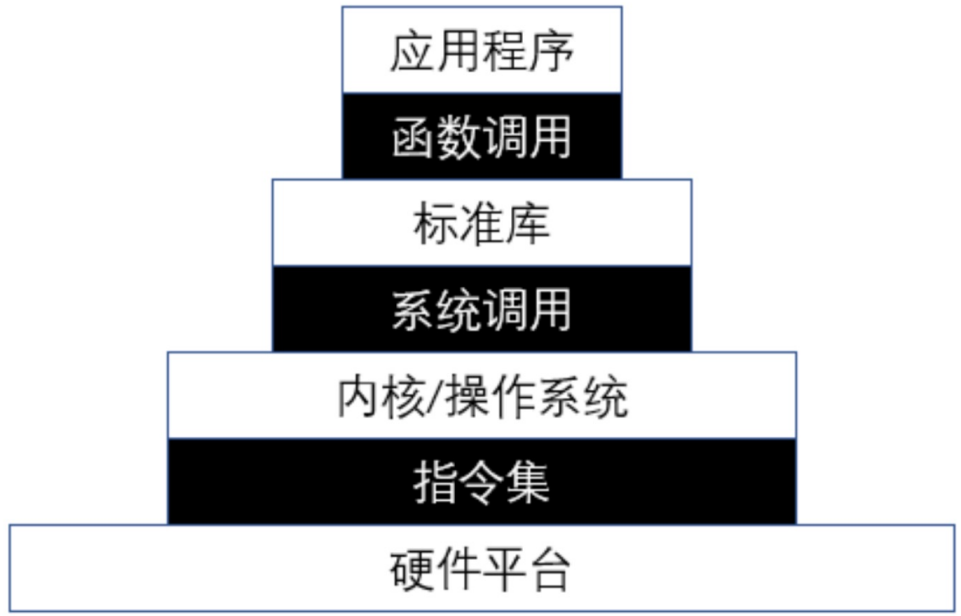


Example: FT2232H

- The device allows for configurable I/O drive strength and slew rate, and it incorporates a highly integrated design with a built-in +1.8V LDO regulator for VCORE, power-on-reset (POR) function, and an on-chip clock multiplier PLL.

Software for Embedded Systems



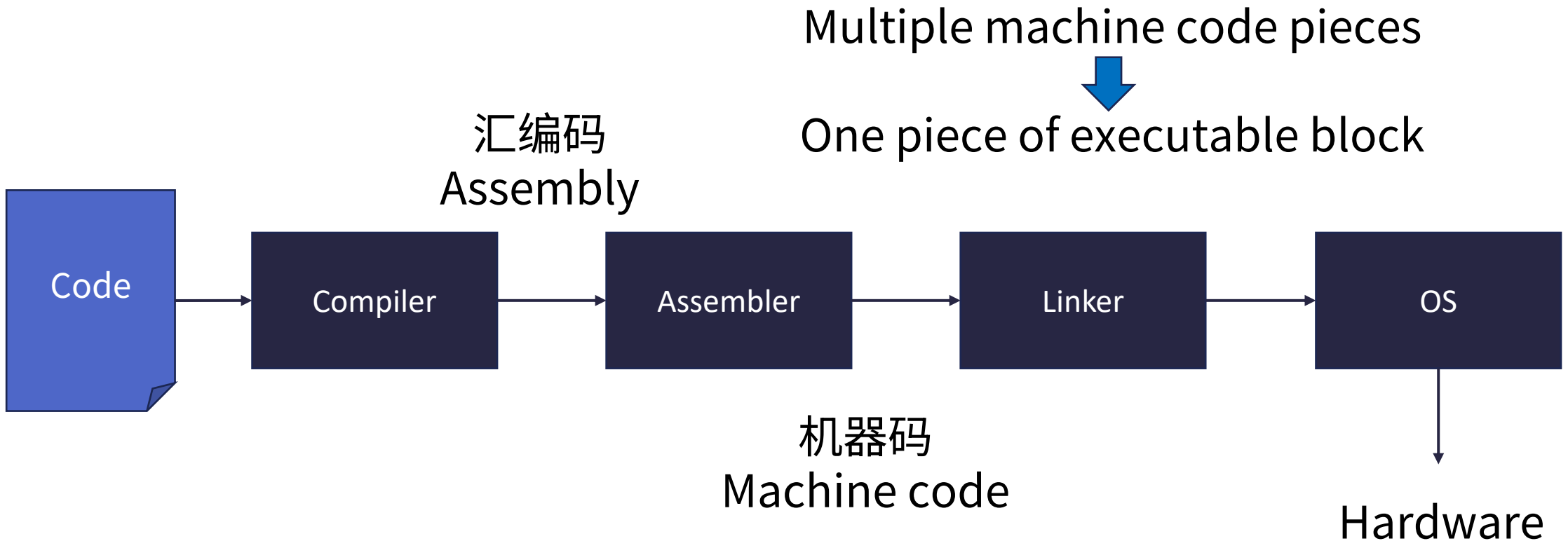


OS: 加载/执行/管理机器码执行文件

Outline

- Introduction to Cyber-Physical System (CPS)
- Introduction to Embedded System
- Embedded System Hierarchy
- Development Flow of Embedded System

Type I - Compilation Work Flow



Type II – MicroPython + Bare Metal Program

- MicroPython was created by Damien George, with initial release launched on May 3, 2014.
- It is actually a program that runs on MicroControllers, and is itself written in C language.

Proper Python with hardware-specific modules

MicroPython is a full Python compiler and runtime that runs on the bare-metal. You get an interactive prompt (the REPL) to execute commands immediately, along with the ability to run and import scripts from the built-in filesystem. The REPL has history, tab completion, auto-indent and paste mode for a great user experience.

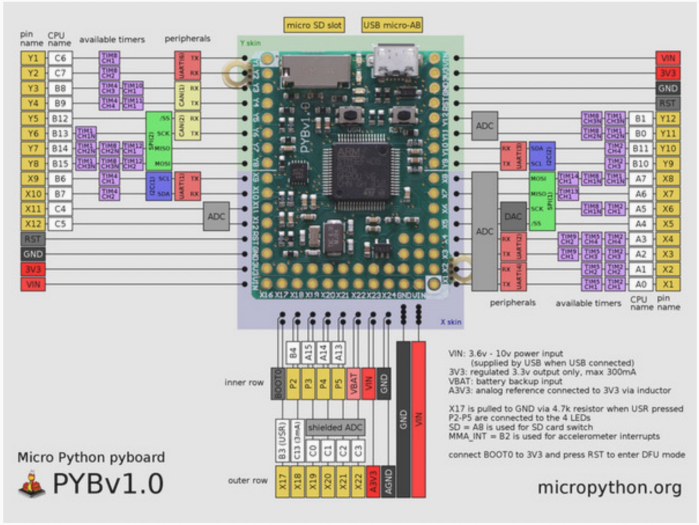
MicroPython strives to be as compatible as possible with normal Python (known as CPython) so that if you know Python you already know MicroPython. On the other hand, the more you learn about MicroPython the better you become at Python.

In addition to implementing a selection of core Python libraries, MicroPython includes modules such as "machine" for accessing low-level hardware.

```
from machine import Pin

# create an I/O pin in output mode
p = Pin('X1', Pin.OUT)

# toggle the pin
p.high()
p.low()
```

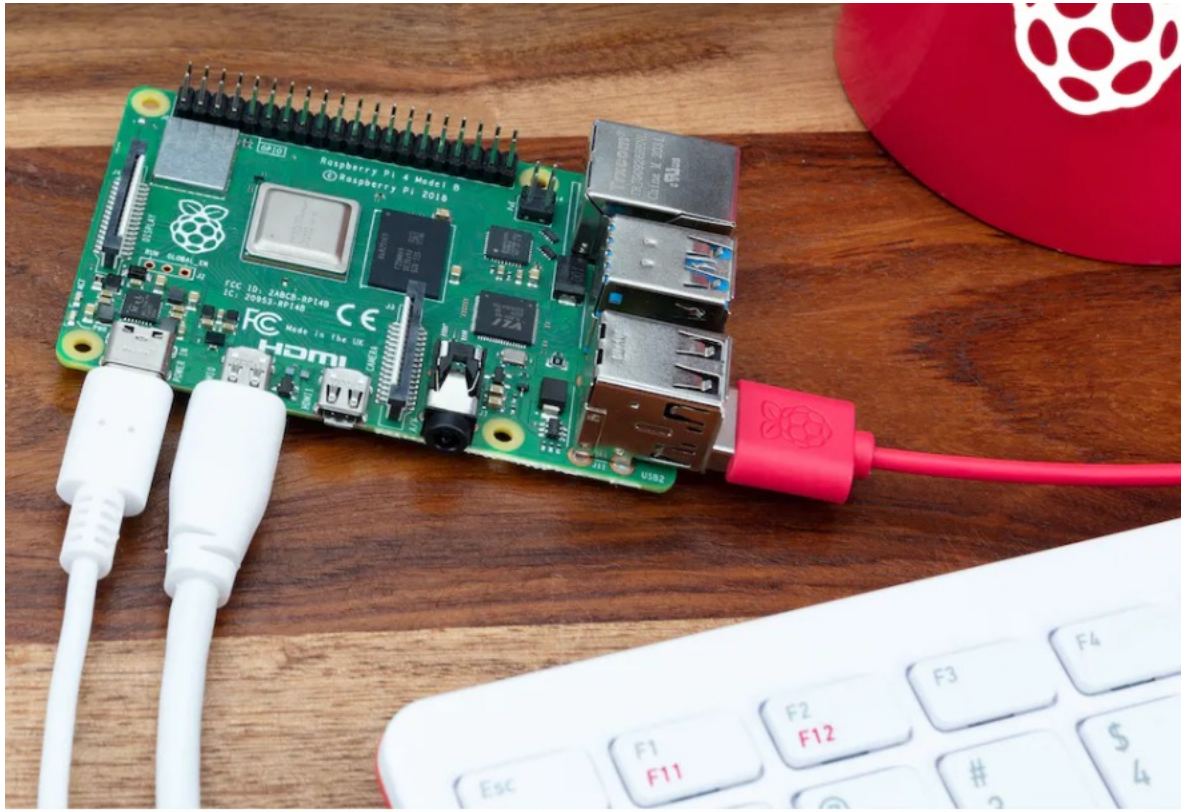


Type III – OS + Python

Raspberry Pi OS

Your Raspberry Pi needs an operating system to work. This is it. Raspberry Pi OS (previously called Raspbian) is our official supported operating system.

Based on Debian Linux distribution

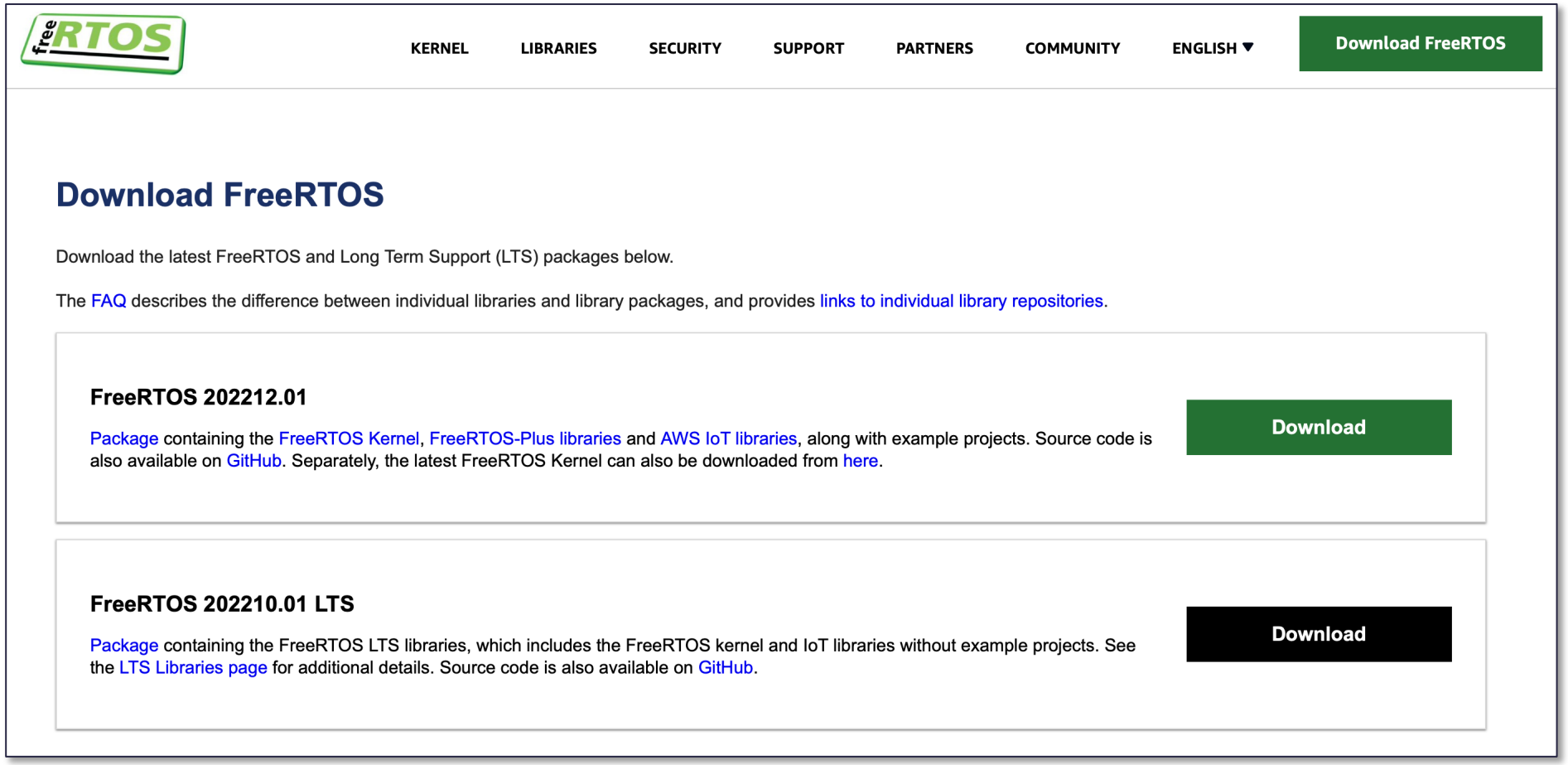


Software for Embedded Systems

火星探测车：勇气号

美国NASA自20世纪80年代末以来，就一直在航天器中采用实时操作系统，从最早的“火星登陆者”到“勇气号”都采用了RTOS（VxWorks)。勇气号采用的嵌入式软件——在仅仅128MB的空间里存放了其最核心的程序

Real-Time Operating System (RTOS) Example

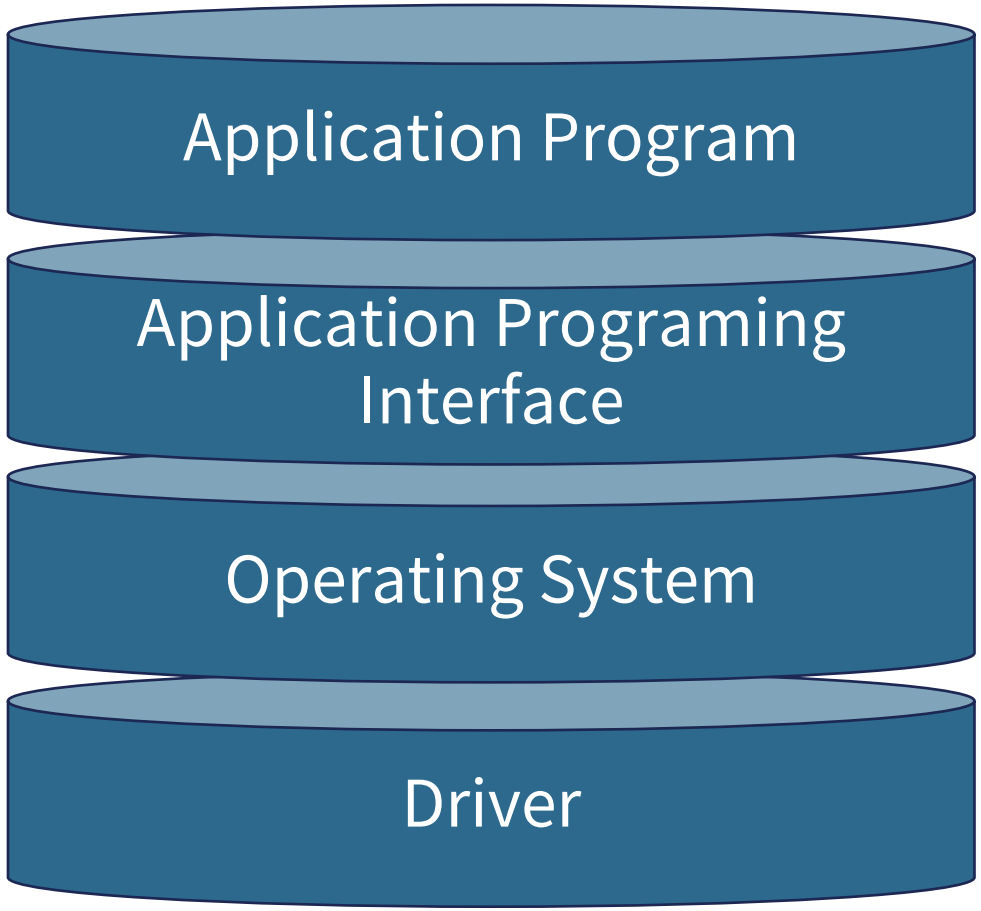


https://www.freertos.org, as of 2023

Real-Time Operating System (RTOS) Feature

Regular OS	Real-Time OS (RTOS)
Complex	Simple
Best effort	Guaranteed response
Fairness	Strict Timing constraints
Average Bandwidth	Minimum and maximum limits
Unknown components	Components are known
Unpredictable behavior	Predictable behavior
Plug and play	RTOS is upgradable

Software for Embedded Systems



NVIDIA

CUDA Runtime API

API Reference Manual

Table of Contents

Development of Embedded Systems

- 开发环境
  - IDE（Intergrated Development Environment）：交叉开发软件一般为一个整合编辑、编译汇编链接、调试、工程管理及函数库等功能模块
    - 在线调试（On-Chip Debugging，OCD）
    - 基于JTAG的ICD（In-Circuit Debugger）
  - 软件模拟环境（如QEMU）
  - 评估电路板（开发板）

Hardware/Software Codesign

- 嵌入式系统设计是使用一组物理硬件和软件来完成所需功能的过程。系统是指任何由硬件、软件或者两者的结合来构成的功能设备。

- 由于嵌入式系统是一个专用的系统，所以嵌入式产品的设计过程中，软件设计和硬件设计是紧密结合、相互协调的。这就产生了一种全新的发展中的设计理论软硬件协同设计

- 这种方法的特点是在设计时，从系统功能的实现考虑，把实现时的软硬件同时考虑进去，硬件设计包括芯片级“功能定制”设计。既可以最大限度的利用有效资源、缩短开发周期，又能取得更好的设计效果。