

第二章 MSP430系列单片机结构

第1节 单片机简介

第2节 MSP430系列单片机结构

第1节 单片机简介

- 1 单片机与嵌入式系统
- 2 典型单片机的基本构成

1. 单片机与嵌入式系统

计算机按应用分类

❖ 通用计算机系统：“非嵌入式应用”

满足海量高速数值计算，主要用于信息处理，能独立使用的计算机系统。

如：个人计算机，工作站等。

技术要求是高速、海量的数值计算；

技术发展方向是总线速度的提升，存储容量的扩大。



❖ 嵌入式计算机系统：“嵌入式应用”

作为其它系统组成部分、以嵌入的形式“隐藏”在各种装置，产品和系统中，实现嵌入式应用的计算机称为嵌入式计算机系统，简称嵌入式系统(Embedded System)。



嵌入式系统(Embedded System)

定义

- IEEE的定义: They are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system.
- 国内普遍认可的定义: 以应用为中心, 以计算机技术为基础, 软件硬件可裁剪, 符合应用系统对功能、可靠性、成本、体积、功耗等严格要求的专用计算机系统。

在嵌入式系统的概念广泛使用后,为了强调单片机的嵌入属性,也称其为嵌入式微控制器。

单片机是典型的嵌入式系统。

汽车电子

平均每辆轿车内使用30~100个MCU
30%以上汽车成本花在汽车电子上

信息系统

信息处理
娱乐

GPS导航系统
移动通信

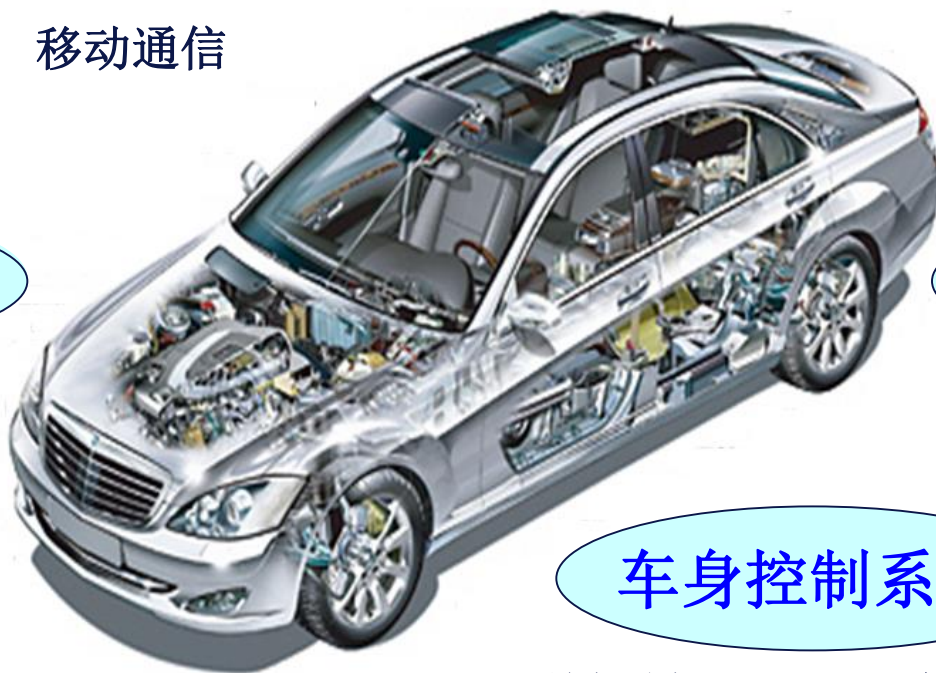
传动控制系统

发动机控制
节能控制

车速控制
转向控制

行驶控制系统

仪表盘控制
空调控制
底盘控制



安全控制系统

安全气囊
防滑控制

车身控制系统

开关控制
门窗控制

车灯控制
防盗控制

无人驾驶汽车： 高精度、高速度、高复杂度的嵌入式控制系统

无人驾驶汽车的工作原理

激光测距仪

能够及时精确地绘制出周边200米之内的3D地形图并上传至车载电脑中枢

车载雷达

微形传感器

负责监控车辆是否偏离了GPS导航仪所制定的路线

电脑资料库

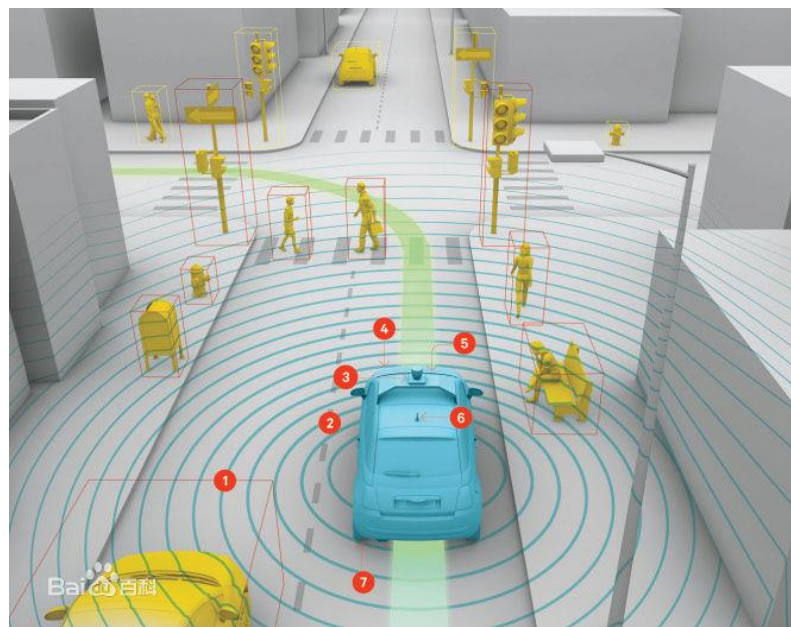
精确地贮存了每条公路的限速标准以及出入口位置,如果处于一名司机的操控下,中央处理系统还会通过扬声器,以柔和悦耳的女声发出类似于“接近十字路口,小心行人”的提示

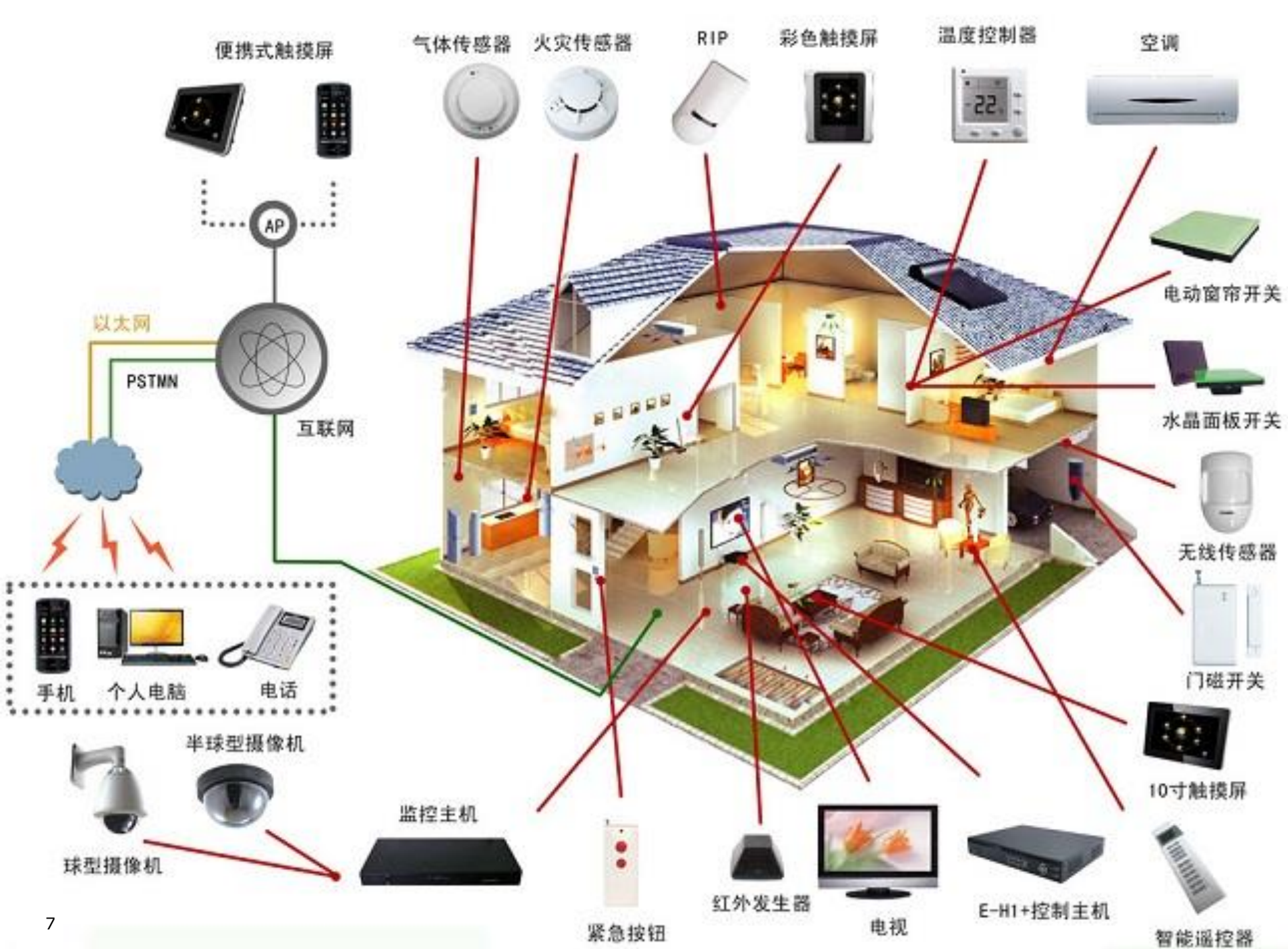
视频摄像头

用以侦测交通信号灯,以及行人、自行车骑行者等车辆行驶路线上遭遇的移动障碍

4个标准车载雷达

以三前一后的布局分布,负责监控较远处的固定路障

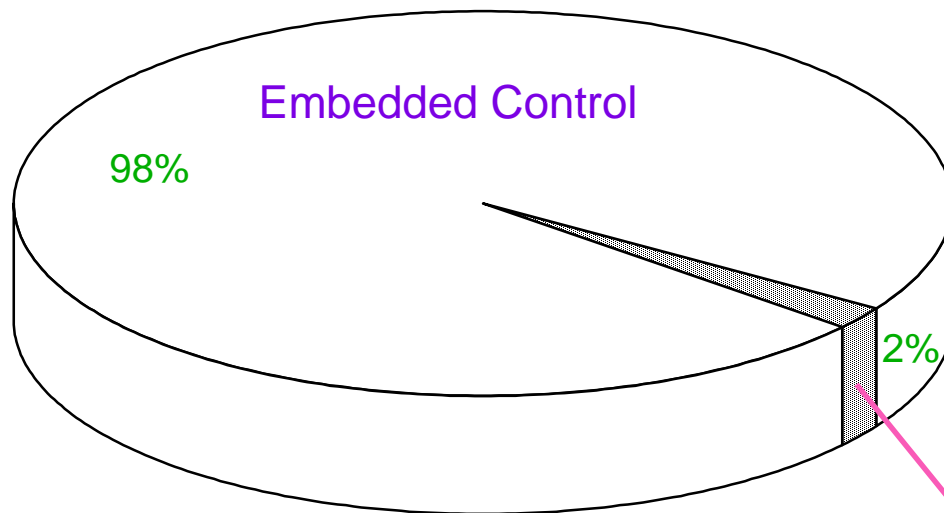






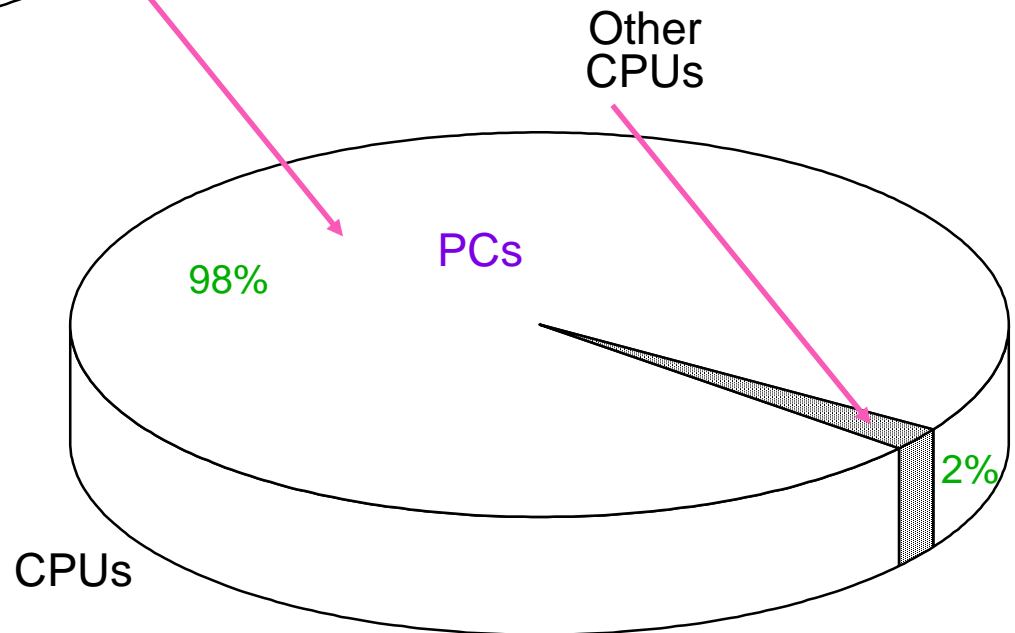
一台通用计算机的外部设备中就包含了多个微控制器：
键盘、鼠标、硬盘、
CD-ROM驱动、
显卡、显示器、
打印机、扫描仪等
均含有MCU。

微处理器应用



All
Microprocessors

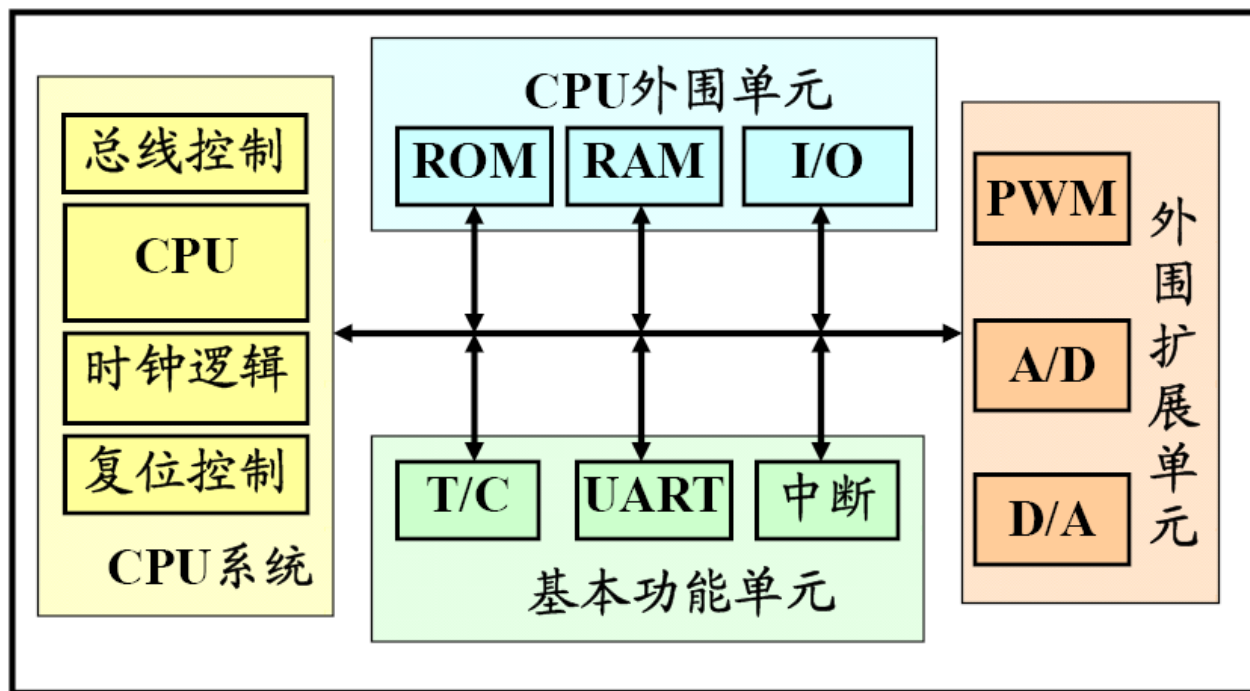
❖ 绝大多数微处理器用于嵌入式场合



CPUs

典型单片机的基本构成

- ◆ 将**CPU**、内存、**I/O**接口集成在一块芯片上，构成具备基本功能的计算机，称**单片机**。



单片机 (Single-chip Microcomputer)

- 超小型、高可靠性、价廉、开发周期短
- 主要应用于控制领域，用以实现各种参数的测量和控制功能，
- 为了强调其控制属性，称其为微控制器 (MCU, MicroController Unit)

单片机的主要应用领域

❖ 工业控制

- 电机控制、物理量的检测与处理、机器人、过程控制、智能传感器、数据传送等。

❖ 仪器仪表

- 智能仪器仪表、医疗器械、色谱仪、示波器等。

❖ 家用电子电器设备

- 电子字典、游戏机、录像机、电冰箱、洗衣机、照相机、空调、防盗控制等。

❖ 通信

- 调制解调器、程控电话交换机、遥控、手机等。

❖ 导航控制

- 鱼雷制导控制、智能武器装置、导弹控制、航天导航系统等。

❖ 数据处理

- 图形终端、复印机、硬盘驱动器、磁带机、打印机、打字机等。

❖ 汽车电子

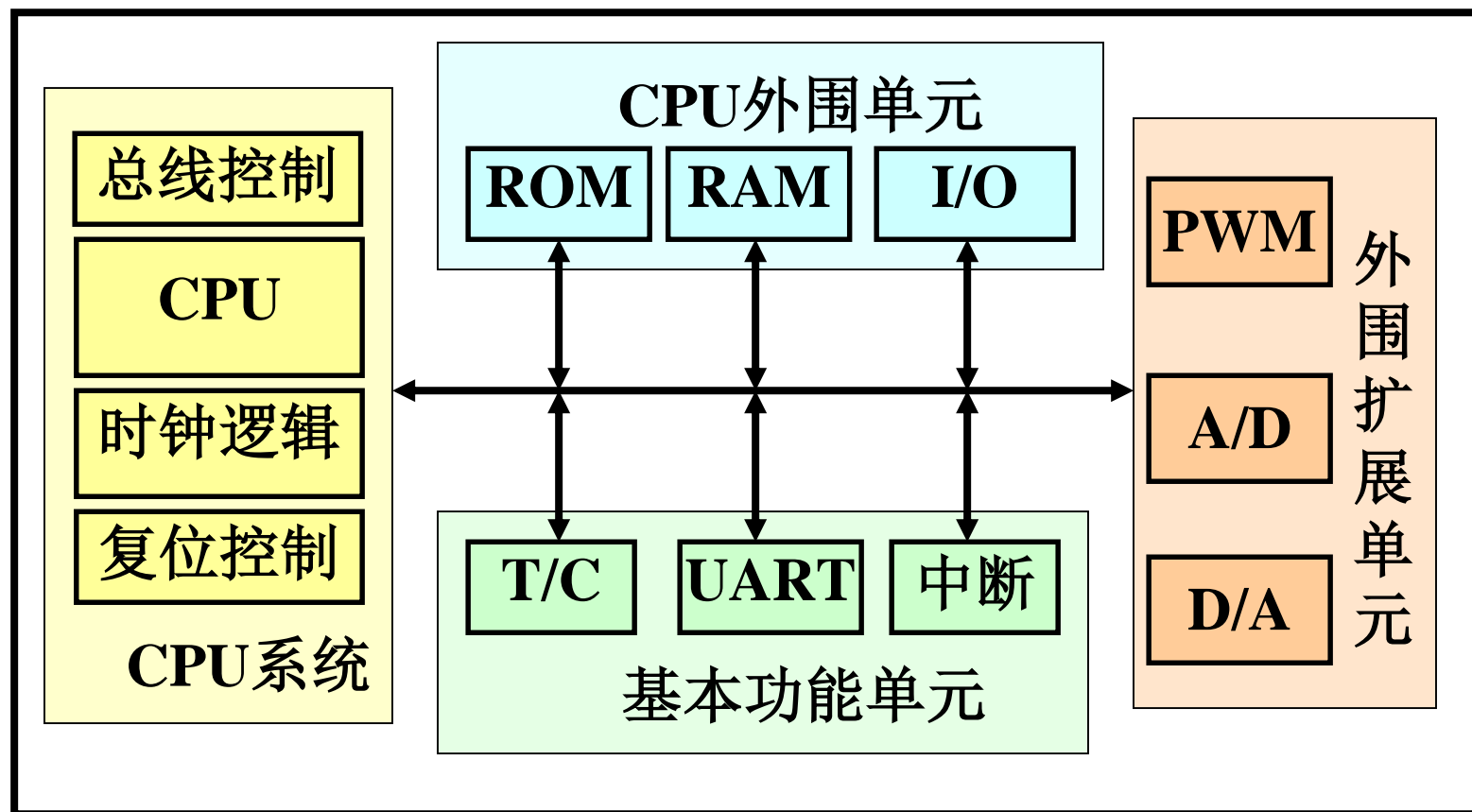
- 点火控制、变速控制、防滑刹车、排气控制、导航、节能控制、冷气控制、报警等

单片机构成的应用系统特点

- ❖ 嵌入到具体的应用系统中，不以计算机的面貌出现；
- ❖ 面向控制对象
 - 面向物理量传感变换的信号输入；
 - 面向对象的伺服驱动控制。
- ❖ 突出控制功能
 - 对外部信息及时捕捉；
 - 对控制对象能灵活地实时控制；
- ❖ 能在工业现场环境中可靠运行。

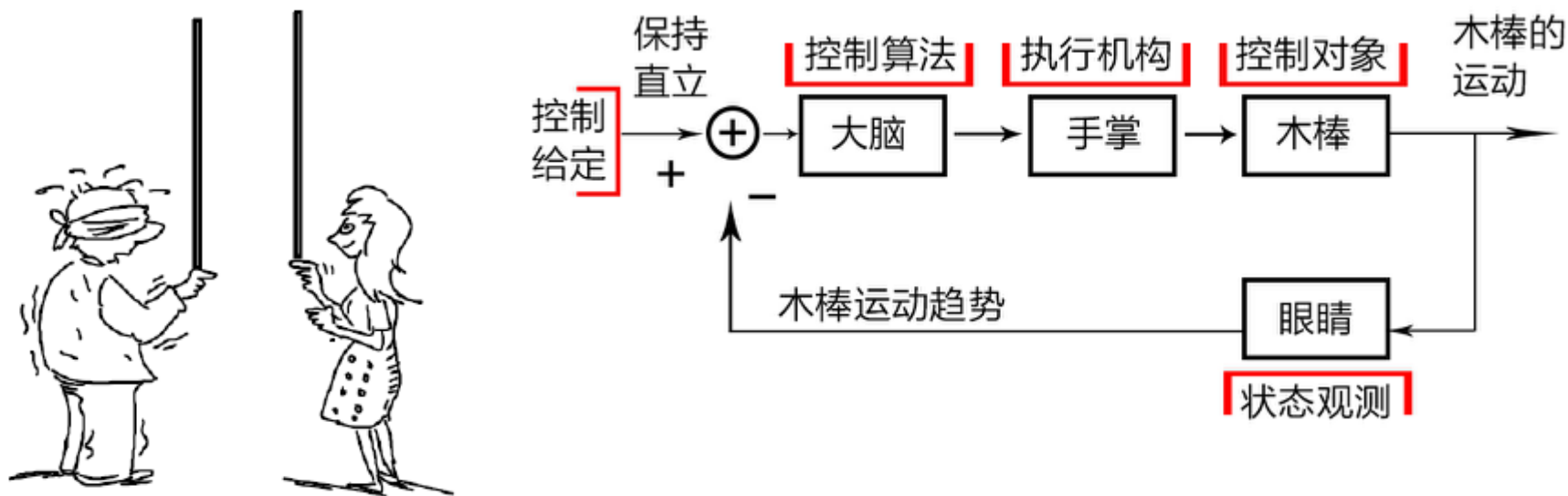
与满足海量高速数值计算的计算机有着明显不同

2. 典型单片机的基本构成



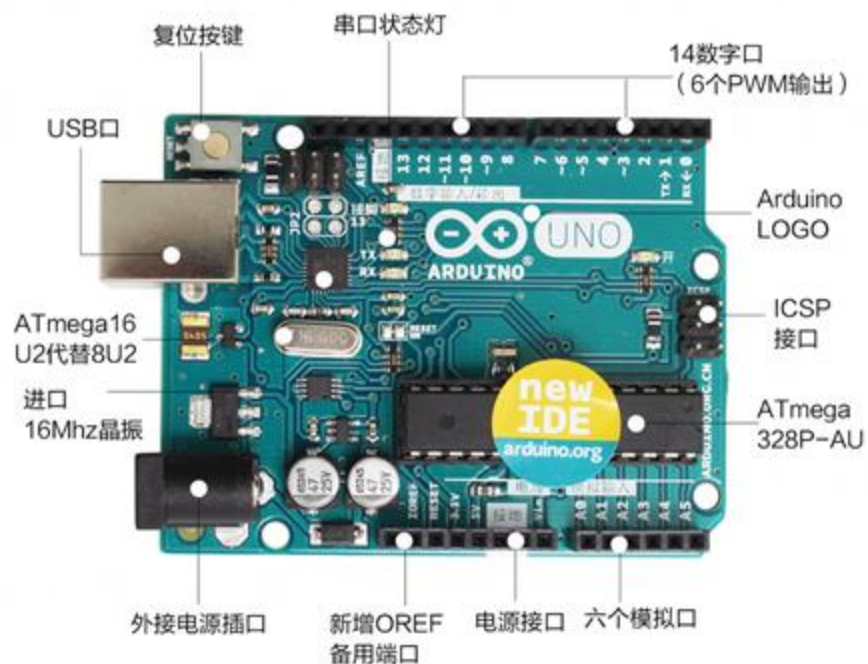
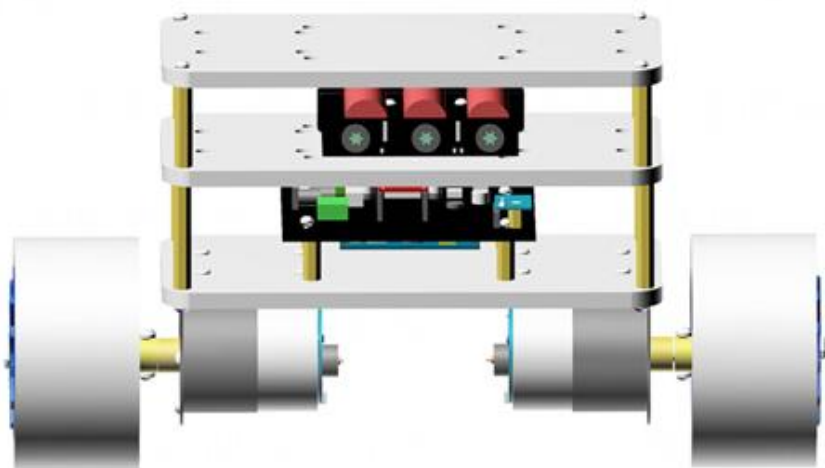
一个生活中控制的实例

一个人是如何控制手上的木棒直立不倒的？



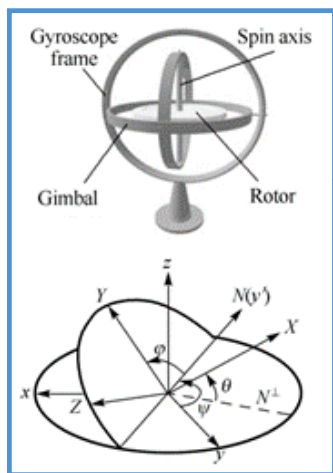
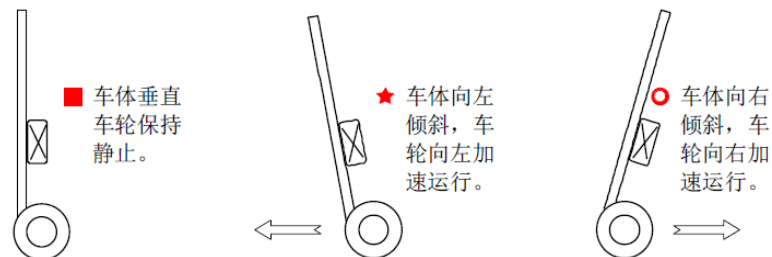
实现这样的控制对微处理器有什么要求？
微处理器的构成是怎么设计的？

单片机应用实例一平衡小车

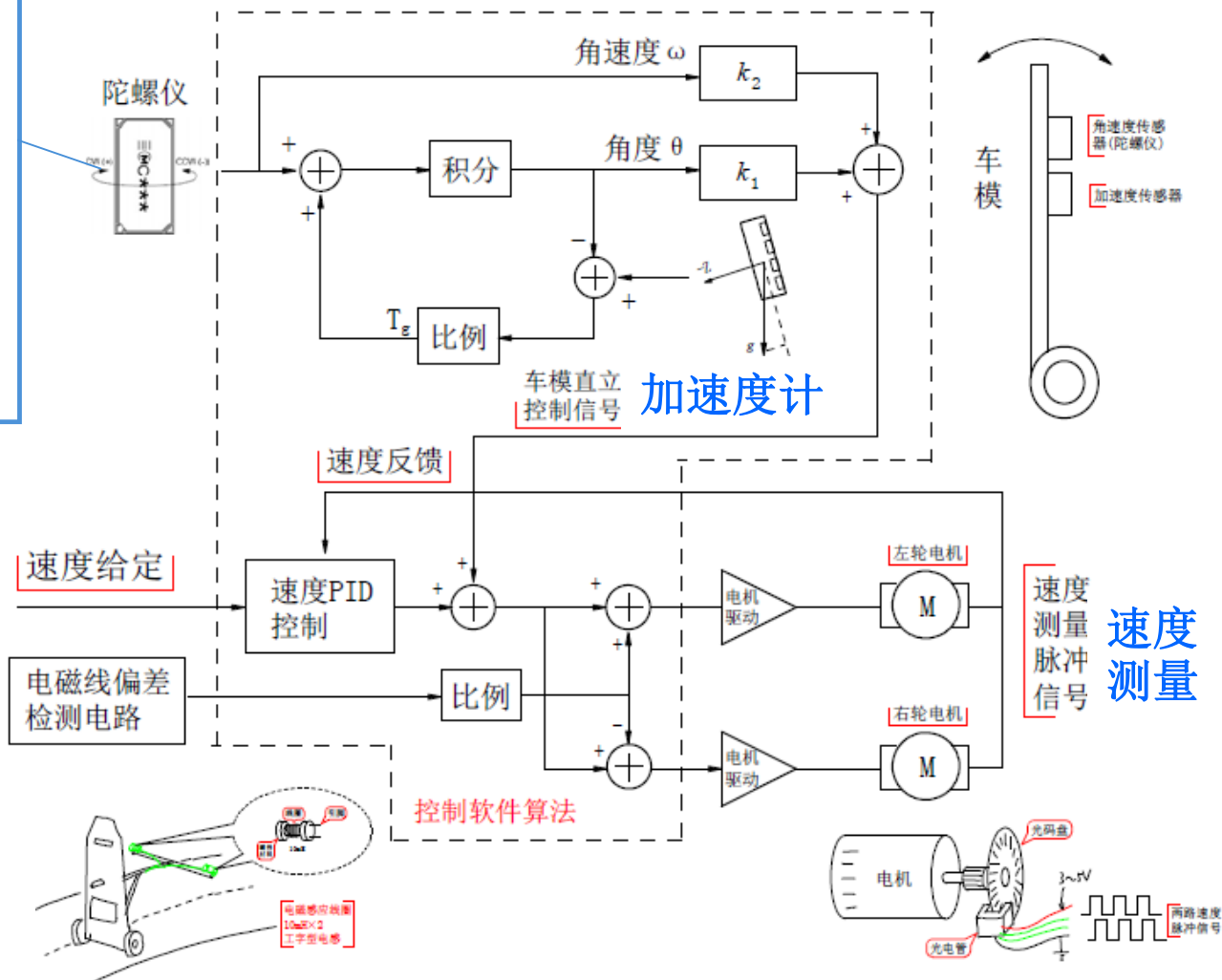


单片机系统实例

平衡小车是怎么控制平衡的？



陀螺仪





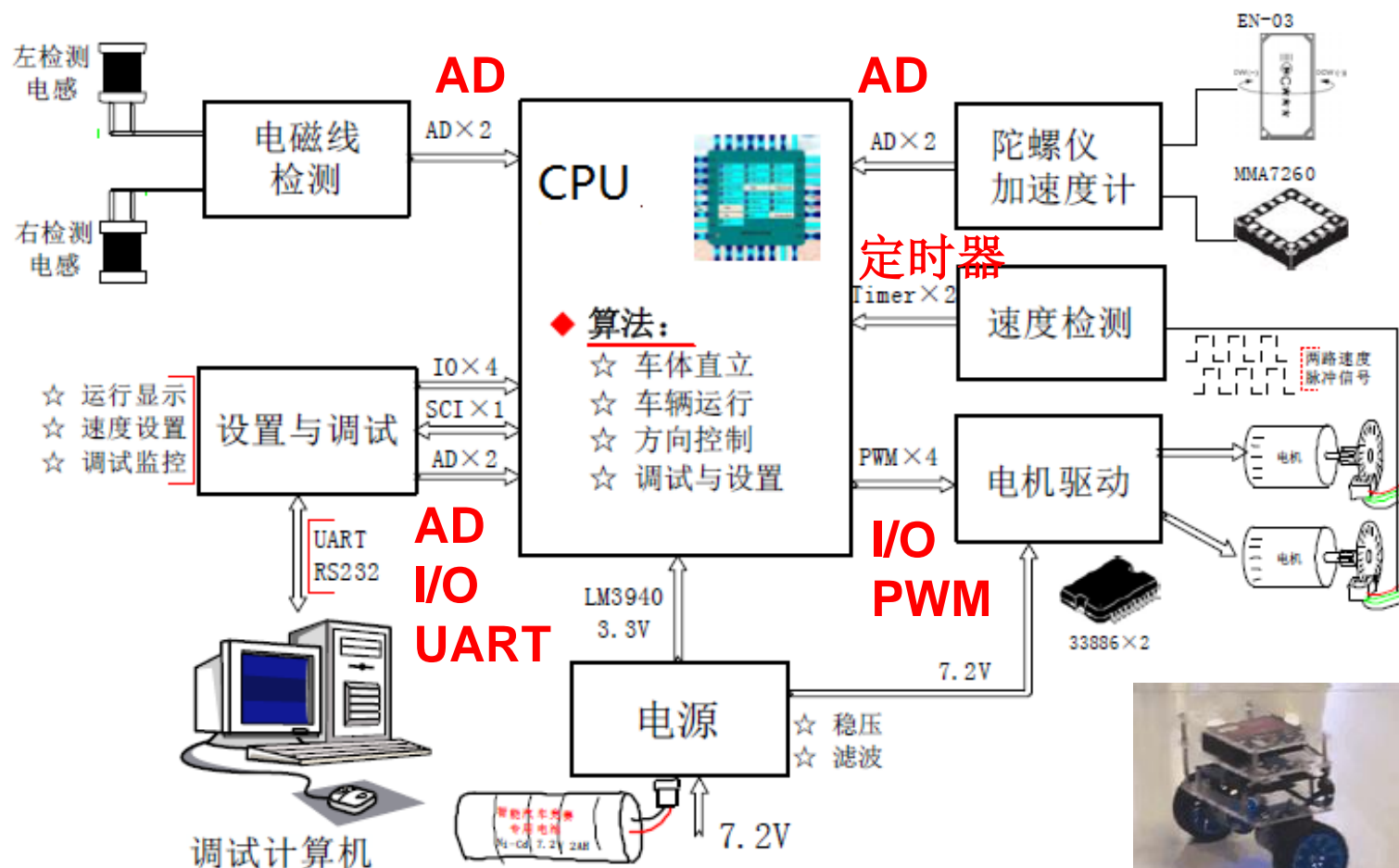
鸟类的头部平衡

虚拟现实游戏
的重力感应

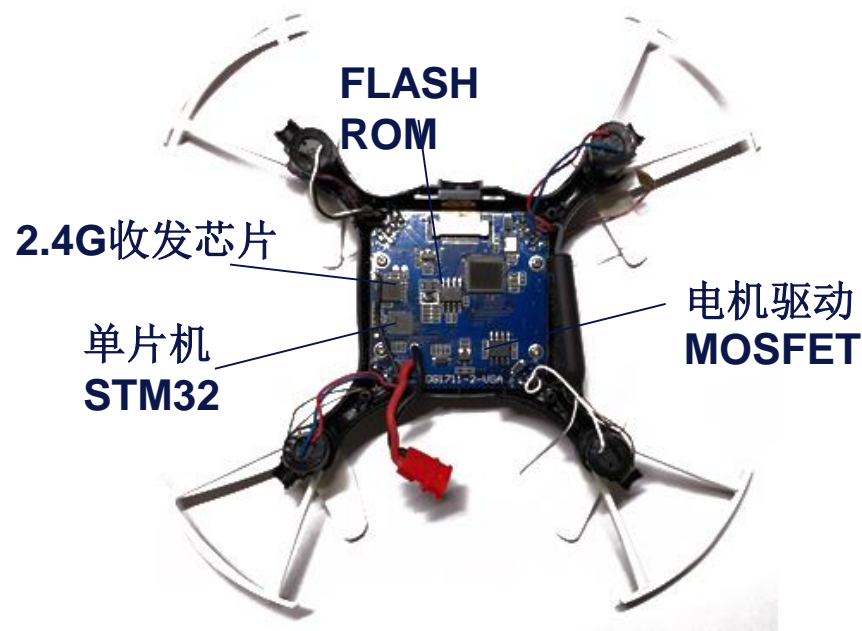
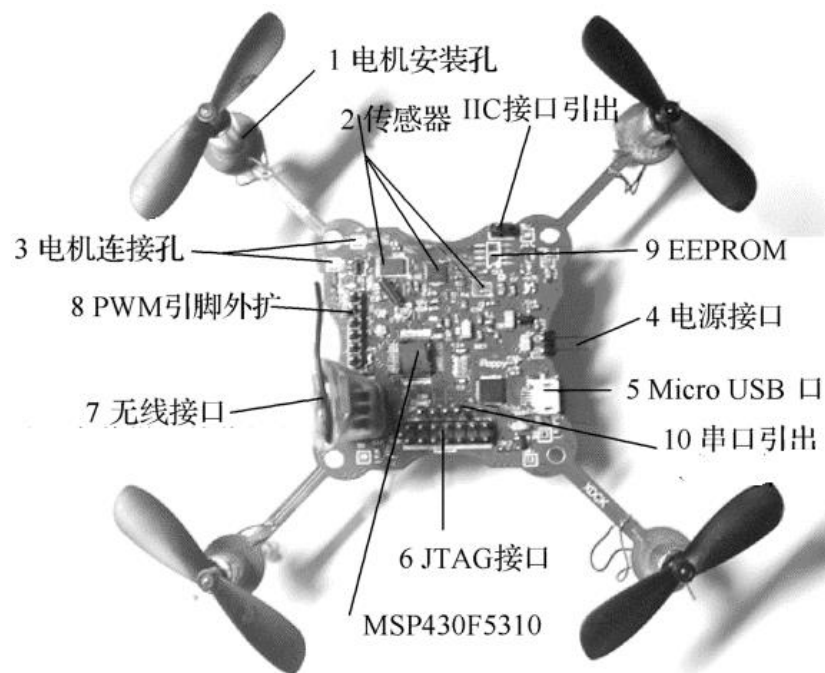


单片机系统实例

平衡车是怎么组成的？



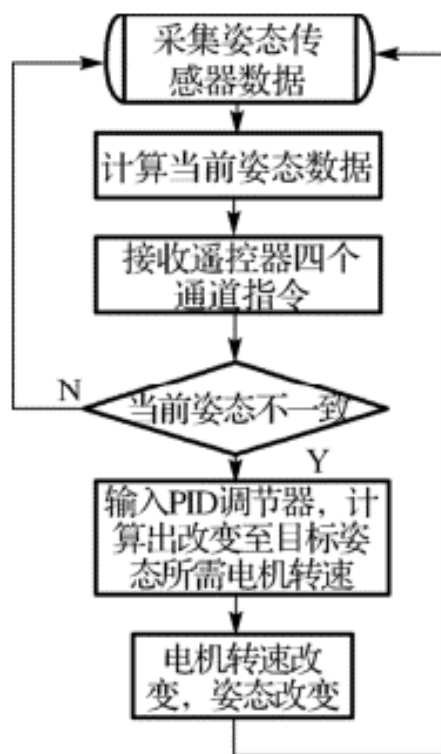
单片机应用实例—四轴飞行器



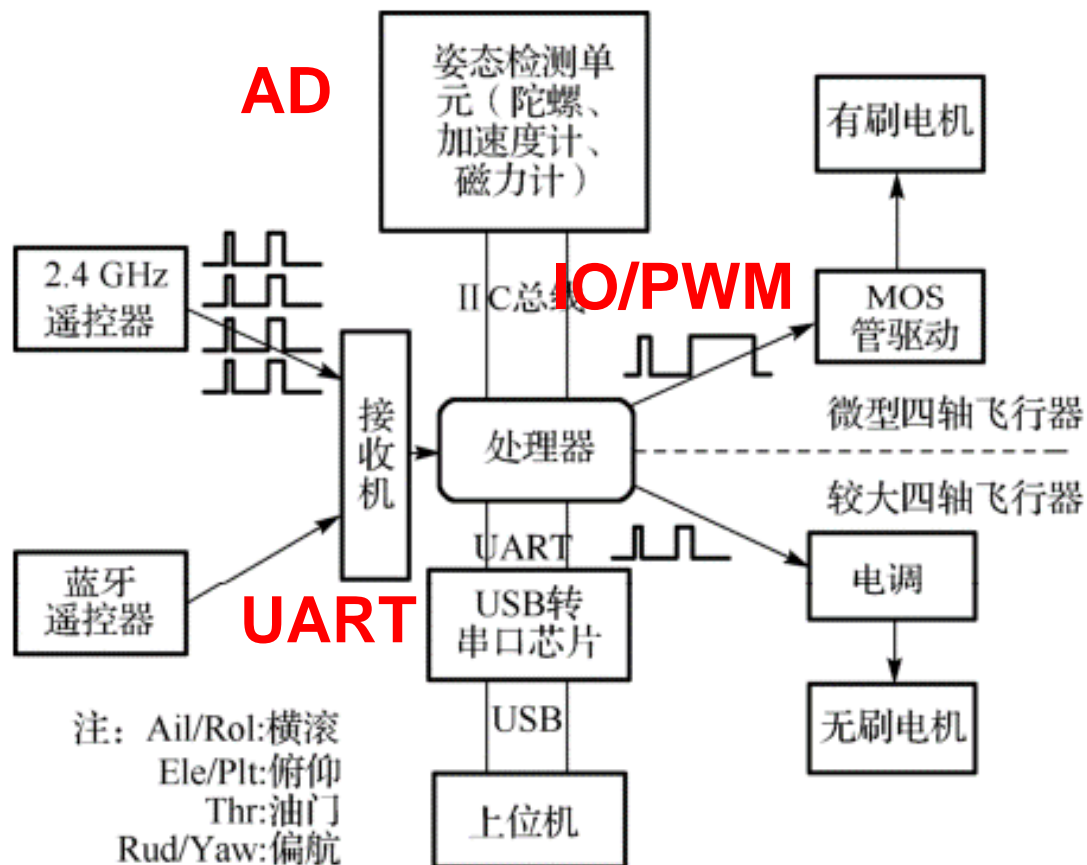
《DIY四轴飞行器：基于MSP430F5系列单片机与Android》

单片机系统实例

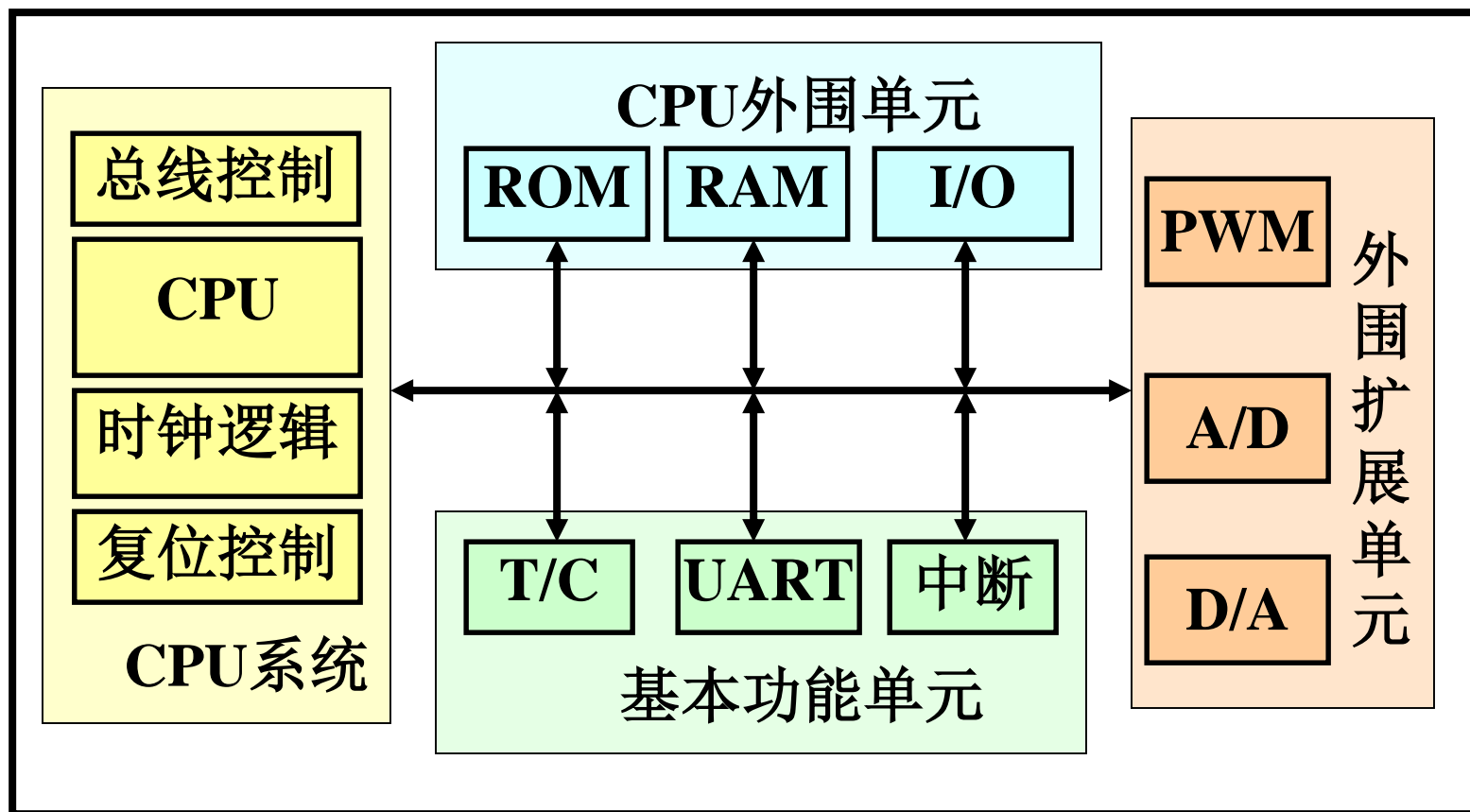
四轴飞行器是怎么控制
平稳飞行的？

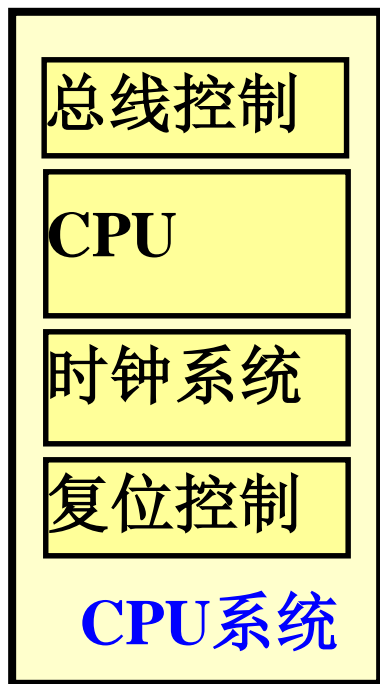


四轴飞行器是怎么组成的？



典型单片机的基本构成





●CPU

内含寄存器运算器，有突出的控制功能。

●时钟系统

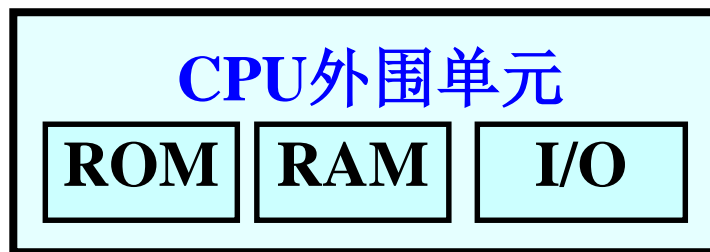
提供各部分的时序同步

●复位电路

提供CPU初始化所需的上电复位信号。

●总线控制逻辑

实现MCU对内部总线和外部总线的控制。

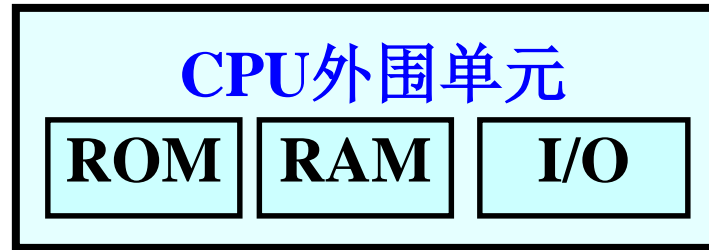


与CPU运行直接相关的单元电路，与CPU构成单片机的最小系统。

●程序存储器**ROM (Read Only Memory)**

用于固化单片机的应用程序和一些表格、常数。

- ①**MaskROM型**
- ②**EPROM型**
- ③**ROMLess型**
- ④**OTPROM型**
- ⑤**FlashROM(MTPROM)型**

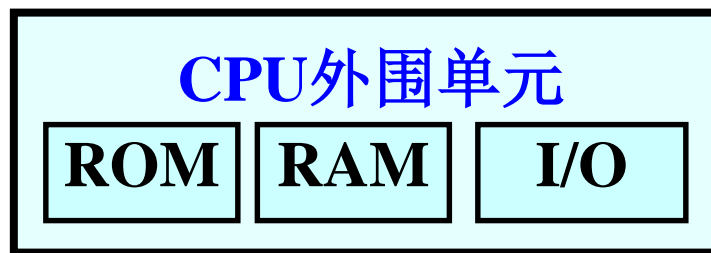


●数据存储器**RAM**

用于临时存放应用程序的处理数据、结果等。

由于面向测控系统，单片机中的数据存储器容量较小，通常是**256字节~2K字节**，

使用静态随机存储器**SRAM(Static Random Access Memory)**。



● I/O端口

单片机与外部的输入输出接口，是芯片的输入/输出引脚。按功能，I/O端口分为以下几种类型：

① 基本I/O端口

用于与外部电路信号的输入/输出控制。

② 片内功能单元的I/O端口

例如，定时器/计数器的计数输入、外部中断源输入等。

③ 总线输入/输出端口

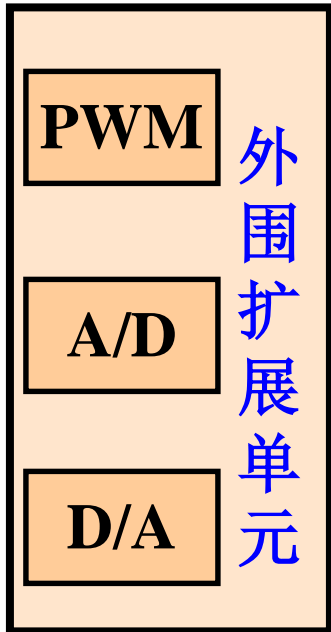
用于外部总线的扩展。

为了减少引脚数量，单片机的I/O端口一般采用复用方式



- 满足单片机**测控功能要求**的基本外围电路，
用来完善和扩大计算机功能。
如定时器 / 计数器、中断系统、串行通信接口等。
- 每个功能单元含一个或多个寄存器，通过编程，
实现这些功能单元的方式设置、启动运行、状态读取等起到对功能单元的运行控制、管理功能。

在有些单片机（**51**系列），称为特殊功能寄存器**SFR**
(**Special Function Register**)

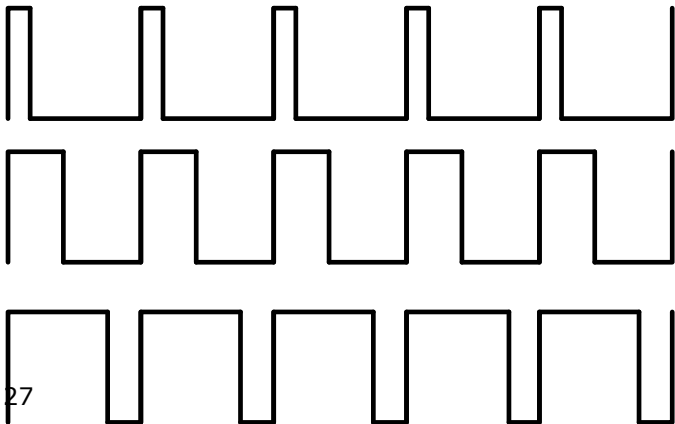


● 满足不同**嵌入式应用要求**的外围扩展功能电路

如 满足数据采集要求而扩展的**ADC, DAC, D/A**

满足伺服驱动控制的**PWM**

满足程序可靠运行的监视定时器**WDT**
(WatchDogTimer) 等



PWM: Pulse Width Modulation

每个系列单片机都有一个CPU基核，
在基核上扩展不同的外围单元电路，
衍生出各种型号的单片机。

例如 TI的MSP430系列MCU
 Intel 的51系列MCU
 Renesas的H8/300H系列MCU

第2节 MSP430系列单片机结构

- 1 MSP430x2xx概述
- 2 MSP430 的CPU编程结构
- 3 MSP430端口的基本输入/输出

1. MSP430x2xx概述

■ MSP430单片机简介

MSP430系列单片机是美国德州仪器公司Ti的16位超低功耗的混合信号处理器(**Mixed Signal Processor**)，针对实际应用需求，把许多模拟电路、数字电路和微处理器集成在一个芯片上，以提供实际应用的“单片”解决方案。

■ MSP430单片机各系列

无段式LCD控制器系列 *MSP430x 3xx、4xx、6xx*

带段式LCD控制器系列 *MSP430 x 1xx、2xx、5xx*



铁电存储器**FRAM**系列

无线射频系列

低电压系列

30 ***MSP430G2**系列价格低廉，经济高效*

从MSP430单片机命名规则了解基本信息



MSP430 **G** **2** **553** **!** **PM**

存储器 ROM类型	
C	ROM
P	OTP
F	FLASH
FR	FRAM
G	FLASH Value line 经济型

各种系列	
1xx	最高8Mhz
2xx	最高16Mhz
3xx	OTP, 已停产
4xx	最高16Mhz 带LCD控制
5xx	最高25Mhz
6xx	最高25Mhz 带LCD控制

器件配置
I/O 端口数
串口类型/个数
ADC类型/个数
DAC类型/个数
段式LCD控制器
硬件乘法器
定时器类型/个数
ROM大小
RAM大小

温度范围	
I	-40~+85°C
T	-40~+105°C

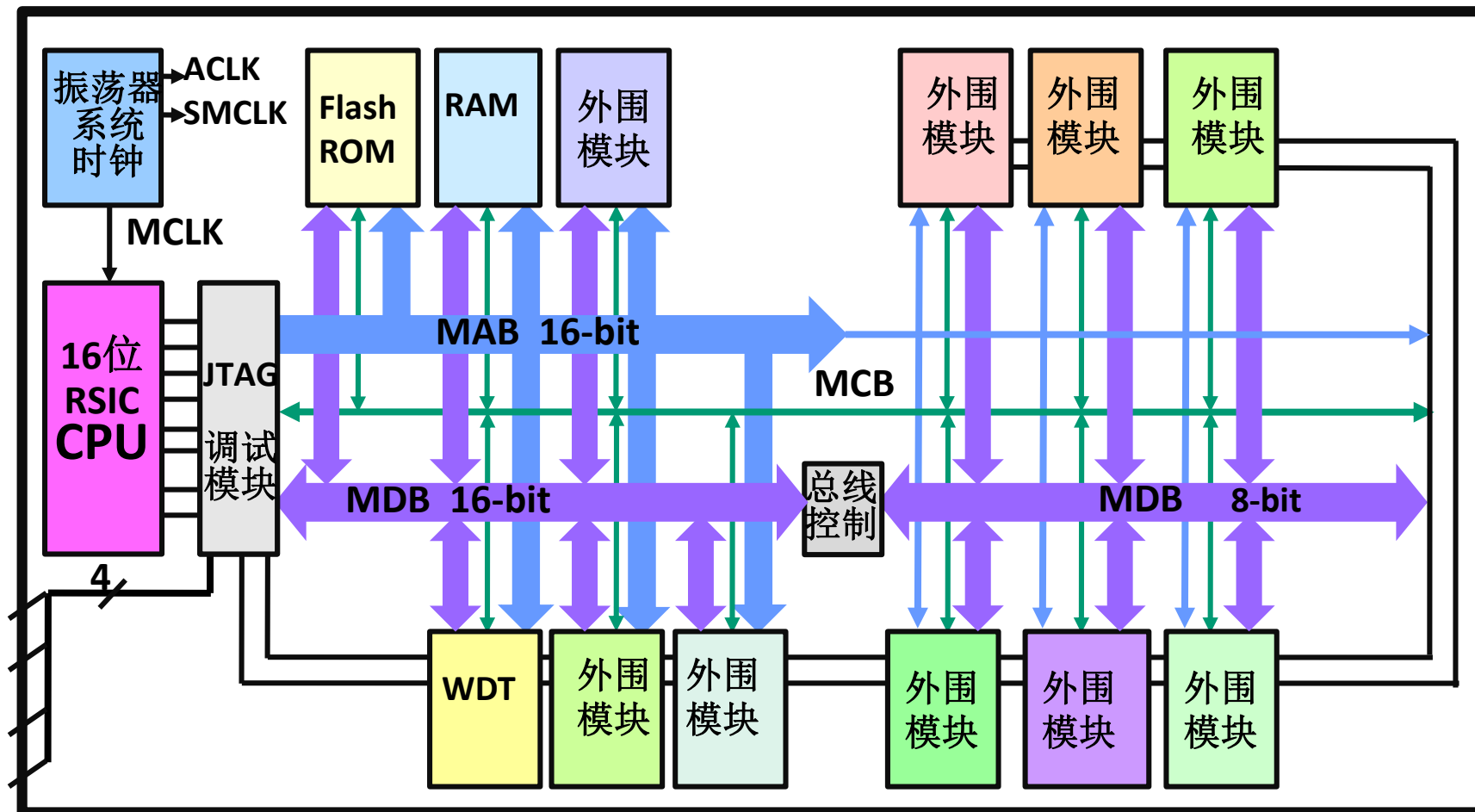
封装类型	
PDIP	双列直插
DW	SOIC20
PM	QFP64 (1mm间距)
PJM	QFP64 (0.5mm间距)

- 用户手册详细描述该系列MCU内部模块功能、使用方法等
如 MSP430**x2xx** Family User Guider.pdf
- MCU数据手册详细描述该MCU内部的配置、引脚功能等
如 MSP430**G2x53**.pdf

MSP430G2xx MCU配置比较

型号	MSP430 G2231	MSP430 G2513	MSP430 G2553
Flash ROM	2KB	16KB	16KB
SRAM	218B	512B	512B
I/O引脚数	10	24	24
看门狗+	有	有	有
TimerA3	1个	2个	2个
USCI A0(UART/SPI)	无	有	有
USCI B0(SPI/I2C)	有	有	有
ADC10	有	无	有
掉电保护	有	有	有

MSP430 结构（框图）



- 各模块通过地址(MAB)、数据(MDB)和控制(MCB)三大总线互连
- 外围模块I/O接口与存储器统一编址

2. MSP430 的CPU编程结构

编程结构

- 指从程序员和使用者的角度看到的结构，与芯片内部的物理结构和实际布局有区别。

编程结构便于程序员从软件角度了解微型计算机系统，而不必关心硬件实现的具体细节。

要点：

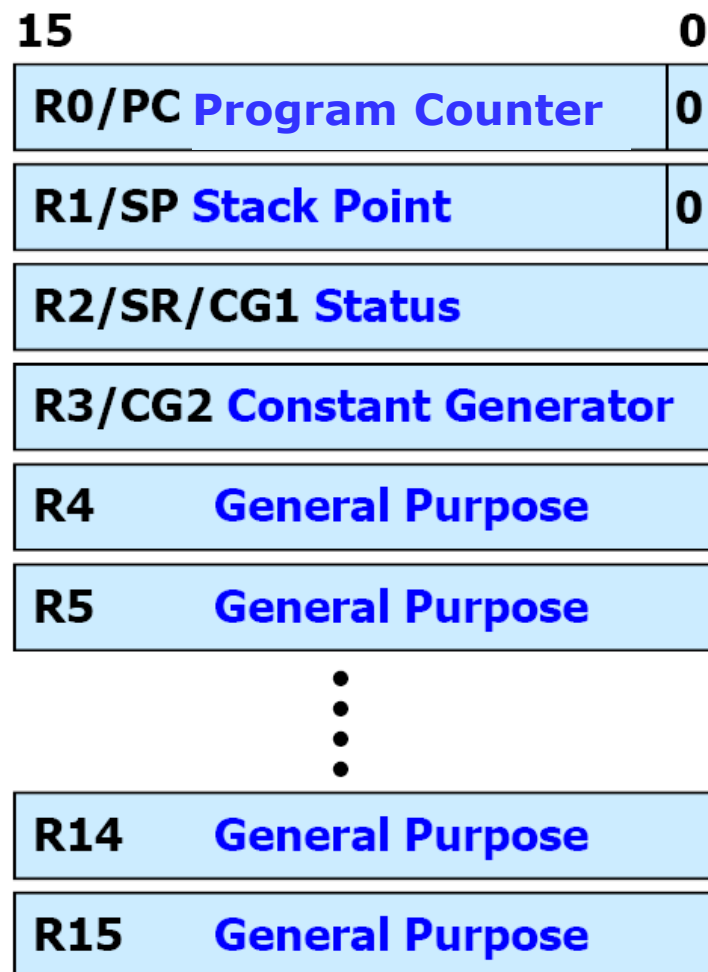
1. 理解CPU内部寄存器
2. 理解存储结构、编址方式
3. 掌握外围单元寄存器的使用方法

MSP430 CPU编程结构要点之一：寄存器组

可以理解为CPU内部的存储单元

❖ MSP430 CPU内含16个16位寄存器

- R0=PC 指令计数寄存器
- R1=SP 堆栈指针寄存器
- R2=SR/CGR1 状态寄存器
- R3=CGR2 常数发生器
- R4~R15 通用寄存器



➤ 状态寄存器SR (Status Register)

16位寄存器，目前用到其中的9位

存放指令执行过程的状态信息和对CPU工作的控制信息

15~9	8	7	6	5	4	3	2	1	0
保留	V	SCG1	SCG0	OSCOFF	CPUOFF	GIE	N	Z	C

V	溢出标志位 - 加减运算结果超出带符号数表示的范围，置V=1
SCG1	系统时钟发生器1 - SMCLK被关闭时，SCG=1
SCG0	系统时钟发生器0 - DCO数字振荡器被关闭时，SCG0=1
OSCOFF	晶振关闭位 - LFXT1晶振被关闭时，OSCOFF=1
CPUOFF	CPU关闭 - CPU被关闭时，CPUOFF=1
GIE	中断使能位 - 可屏蔽中断使能，置GIE=1
N	符号标志位 - 当字或字节运算的结果为负数时，置N=1
Z	零标志位 - 运算结果为零，置Z=1
C ³⁶	进位标志位 - 运算过程中，最高位产生进位，置C=1

例 8位二进制加法如下，给出各状态标志位的值

$$\begin{array}{r}
 10110101 \\
 + 10001111 \\
 \hline
 \text{进位 } 1 \quad 111111 \\
 01000100
 \end{array}$$

被加数8位
加数8位

和8位

最高位D7位产生进位:

相加的结果为44H, 不为0:

结果的最高位为0:

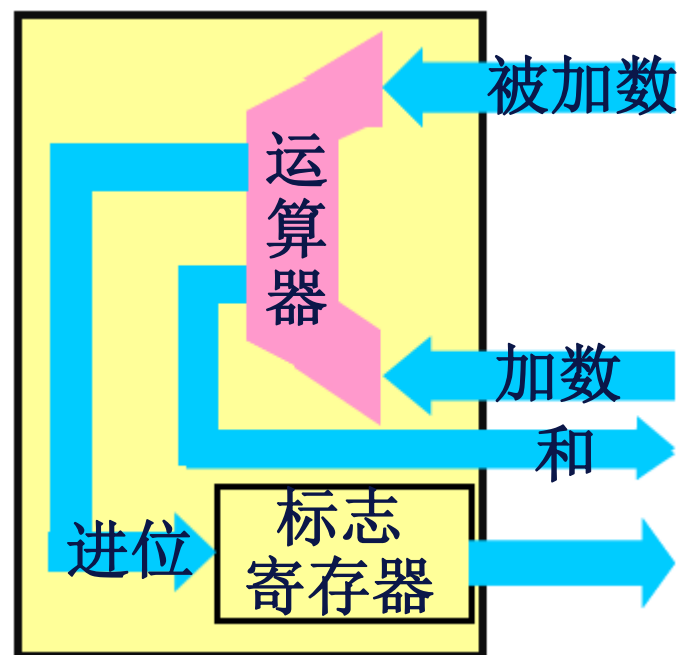
负数+负数=正数:

C = _

Z = _

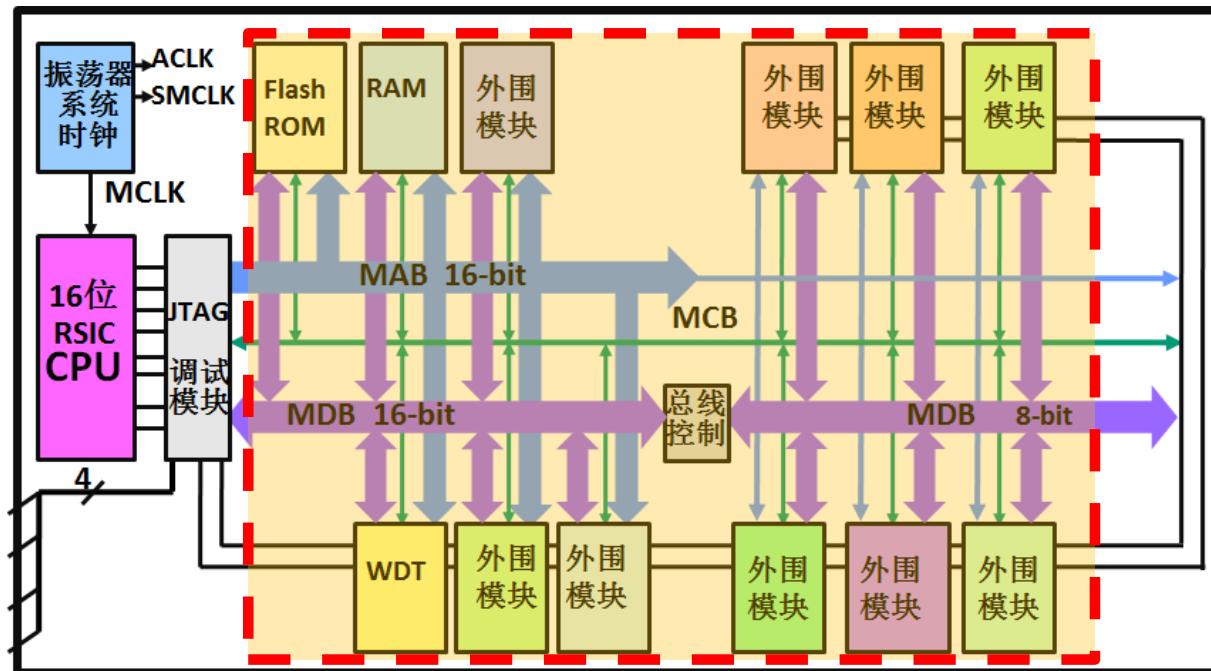
N = _

V = _



15~9	8	7	6	5	4	3	2	1	0
保留	V	SCG1	SCG0	OSCOff	CPUoff	GIE	N	Z	C

MSP430 存储器



地址总线: 16位

数据总线: 16位
8位

统一编址 对CPU外部空间统一编址

16位地址总线能够编多少个地址?

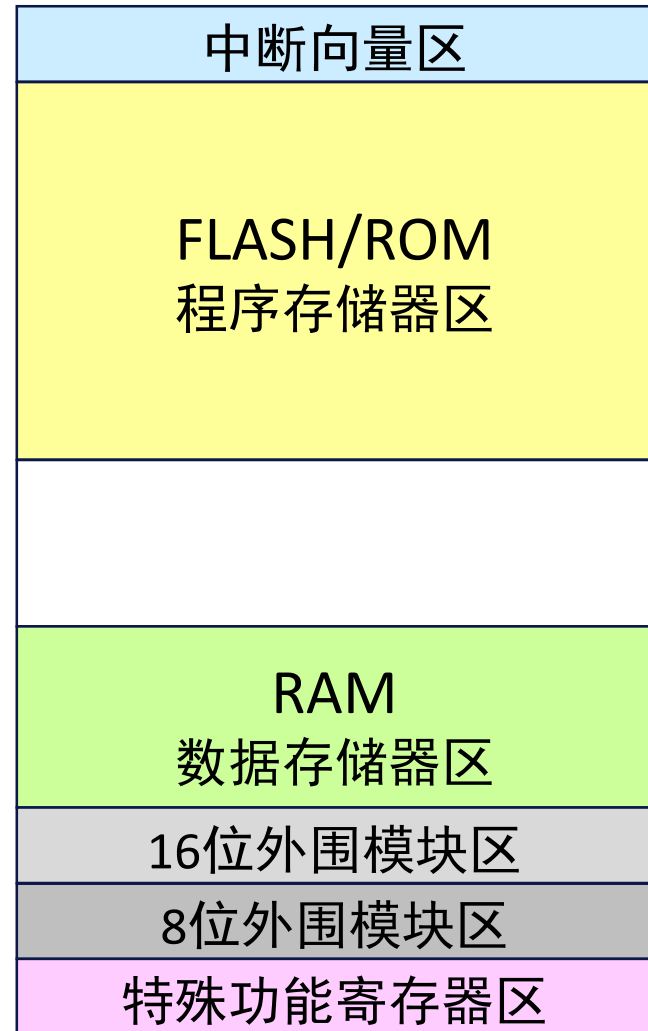
0000h-FFFFh $2^{16}=64k$

地址FFFFH

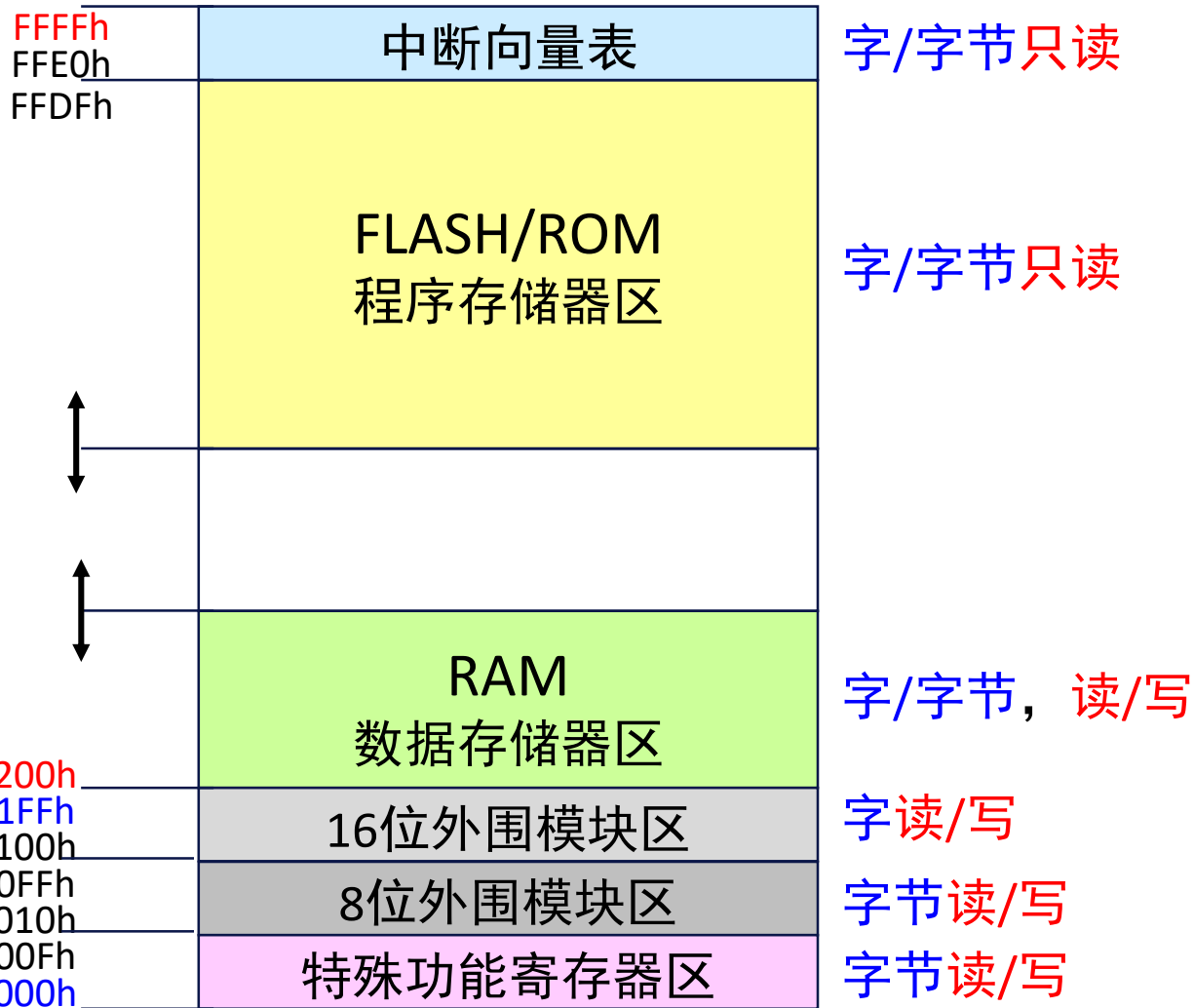
外部空间类似一个园区 CPU类似园区的管理员

- 分为多个大楼（物理上分离，不同功能）
- 每个楼有多个房间（存储单元或寄存器）
- 所有房间统一编房间号（地址）

地址0000H



MSP430的存储器结构



■ 统一编址
 物理上完全分离的
 存储区域
 (如ROM/Flash、RAM、
 外围模块寄存器、
 特殊功能寄存器SFRs等)
 被安排在
 0000~FFFFh范围
 的同一地址空间

- ROM的最高地址都是0FFFFH, 起始地址据MCU型号而不同
- RAM起始地址都是0200H, 末地址据MCU型号而不同;
- 外围模块寄存器都在0~01FFH处

MSP430G2xx3的存储器结构

MSP430		G2153 G2113	G2253 G2213	G2353 G2313	G2453 G2413	G2553 G2513
ROM	容量	1KB	2KB	4KB	8KB	16KB
	中断向量	64B, 0xFFFF~0xFFC0				
	代码存储器	0xFFBF~ x0FC00	0xFFBF~ 0xF800	FFBF~ F000	0xFFBF~ 0xE000	0xFFBF~ 0xC000
	信息存储器	256B, 0x10FF~0x1000				
RAM	容量	256B	256B	256B	512B	512B
	范围	0x02FF~0x0200			0x03FF~0200	
16位外围模块		01FF~0100H				
8位外围模块		00FF~0010H				
特殊功能寄存		000F~0000H				

- 不同型号MCU的存储器ROM和RAM容量不一样,
- ROM的最高地址都是0FFFFH,
- RAM起始地址都是0200H,

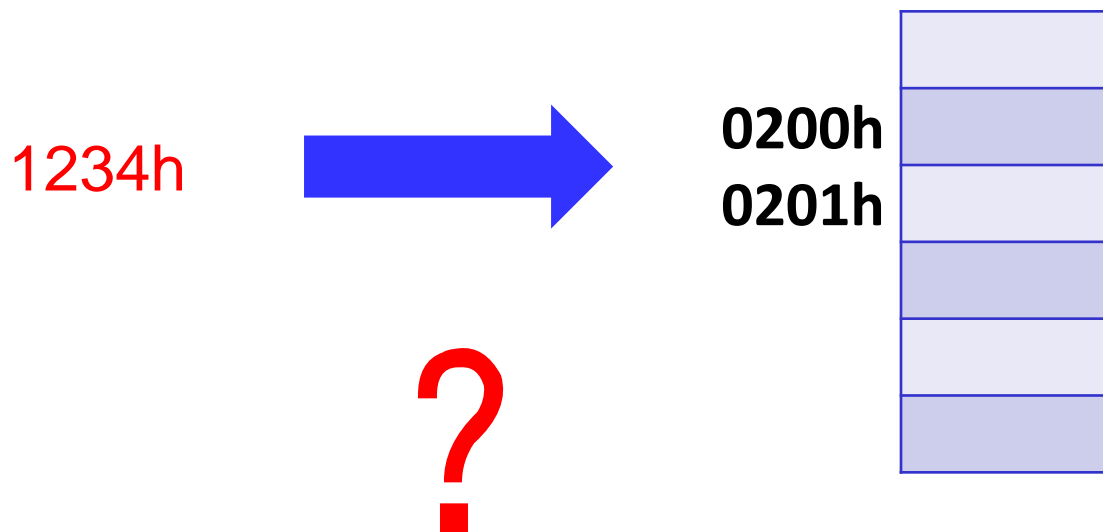
RAM的末地址 = 该器件数据RAM 容量 + 0200H-1;

- 中断向量地址、外围模块地址、特殊功能寄存器地址范围相同

kB是指什么单位？

Byte是8位还是16位的？

16位数据总线，和8位的存储单元如何配合？



MSP430的存储器读写模式

1个存储器单元可存放1个**字节**数据，当往存储器写入1个**字**的数据时，有两种存储器读写模式

➤小端模式 (Little Endian):

数据的低字节存放在内存低地址中，高字节存放在高地址中，即“高高低低”原则

MSP430采用小端模式读写存储器

➤大端模式 (Big Endian):

数据的低字节存放在内存高地址中，高字节存放在低地址中，即“高低低高”原则

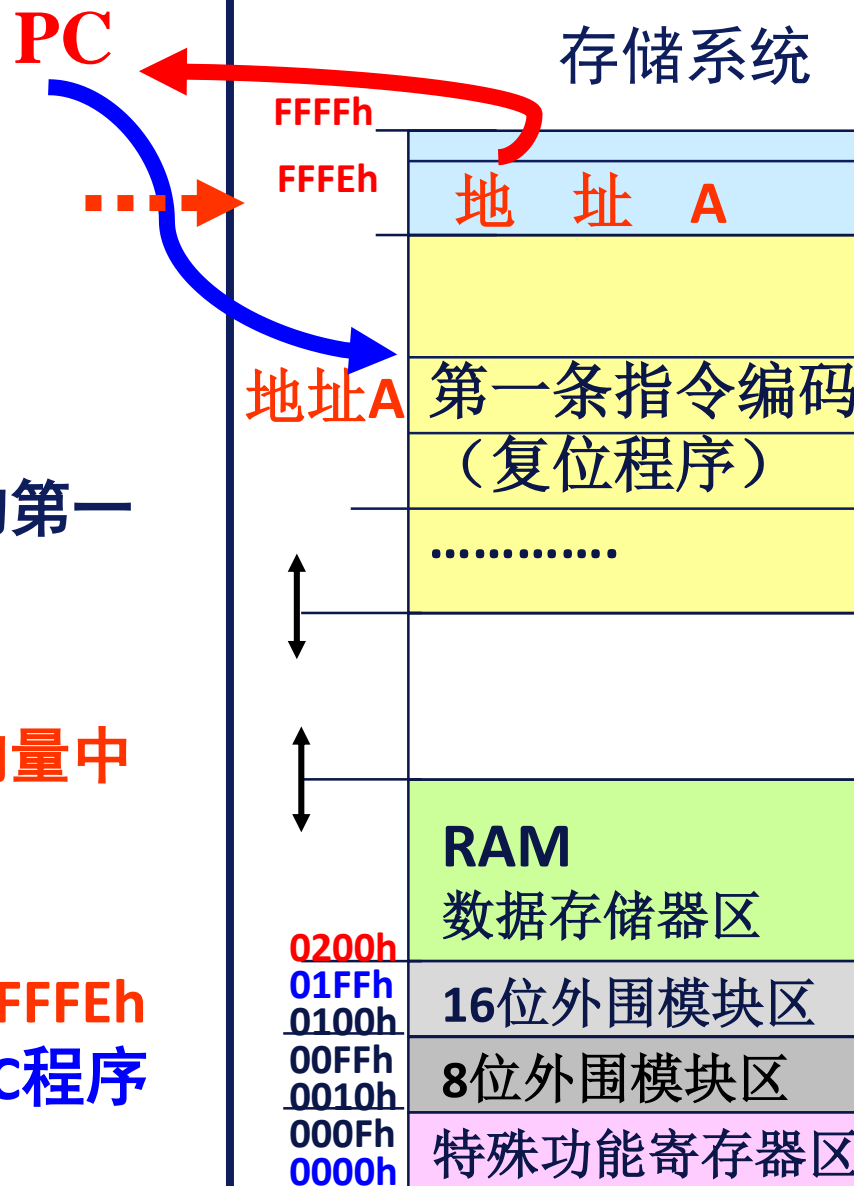
例：将#1234h写入0200h开始的存储单元中

小端模式

0200h	34h
0201h	12h
0202h	
0203h	

大端模式

0200h	12h
0201h	34h
0202h	
0203h	

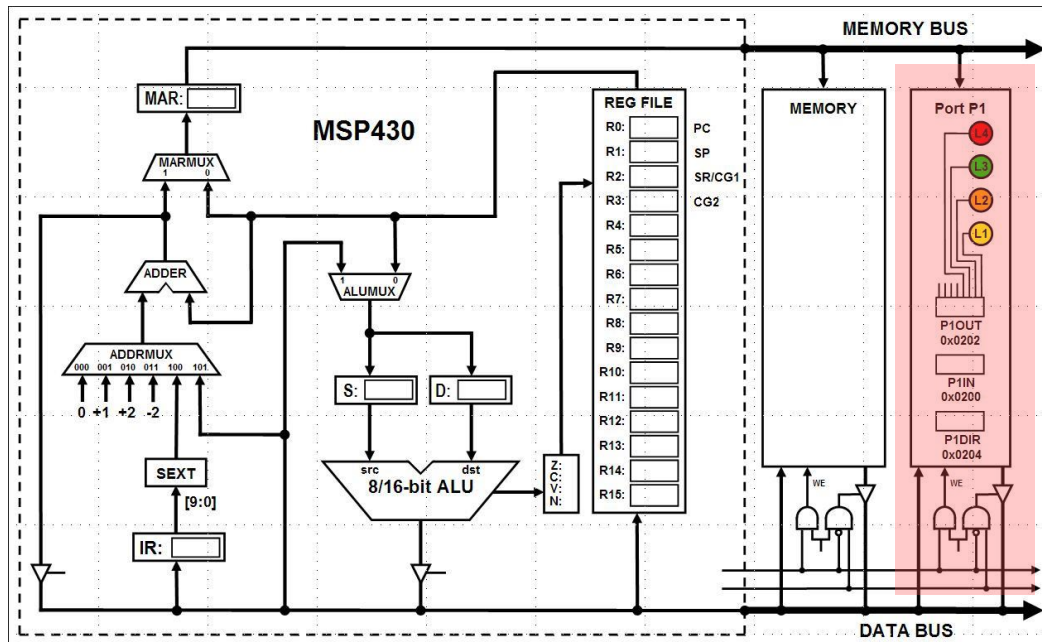
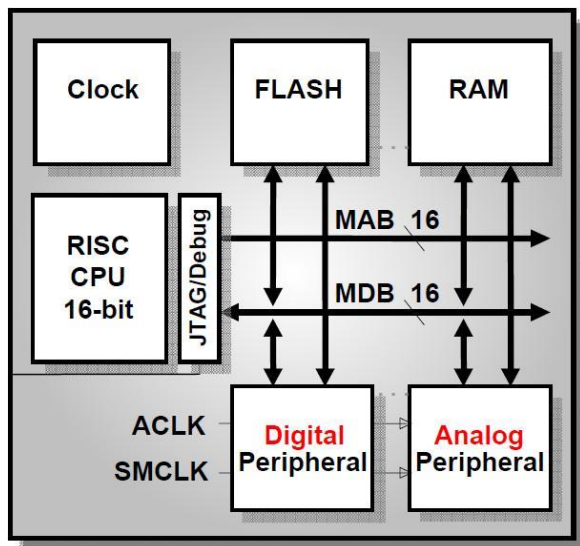


MSP430单片机的上电或复位

通过“复位”，使系统“起死回生”

- 第一条指令：开机或者复位后执行的第一条指令；
- 第一条指令的地址存放于复位中断向量中 (FFFFh和FFFEh 地址中)
- 上电或者复位后，CPU通过FFFFh和FFFEh 存储单元获取第一条指令地址 给 PC 程序计数寄存器，即开始执行存放在那里的程序。

3. MSP430 端口的基本输入/输出(I/O)



- ❖ 外设是人与计算机交互的途径：实现信息输入与输出
- ❖ I/O接口是计算机与外设连接的桥梁

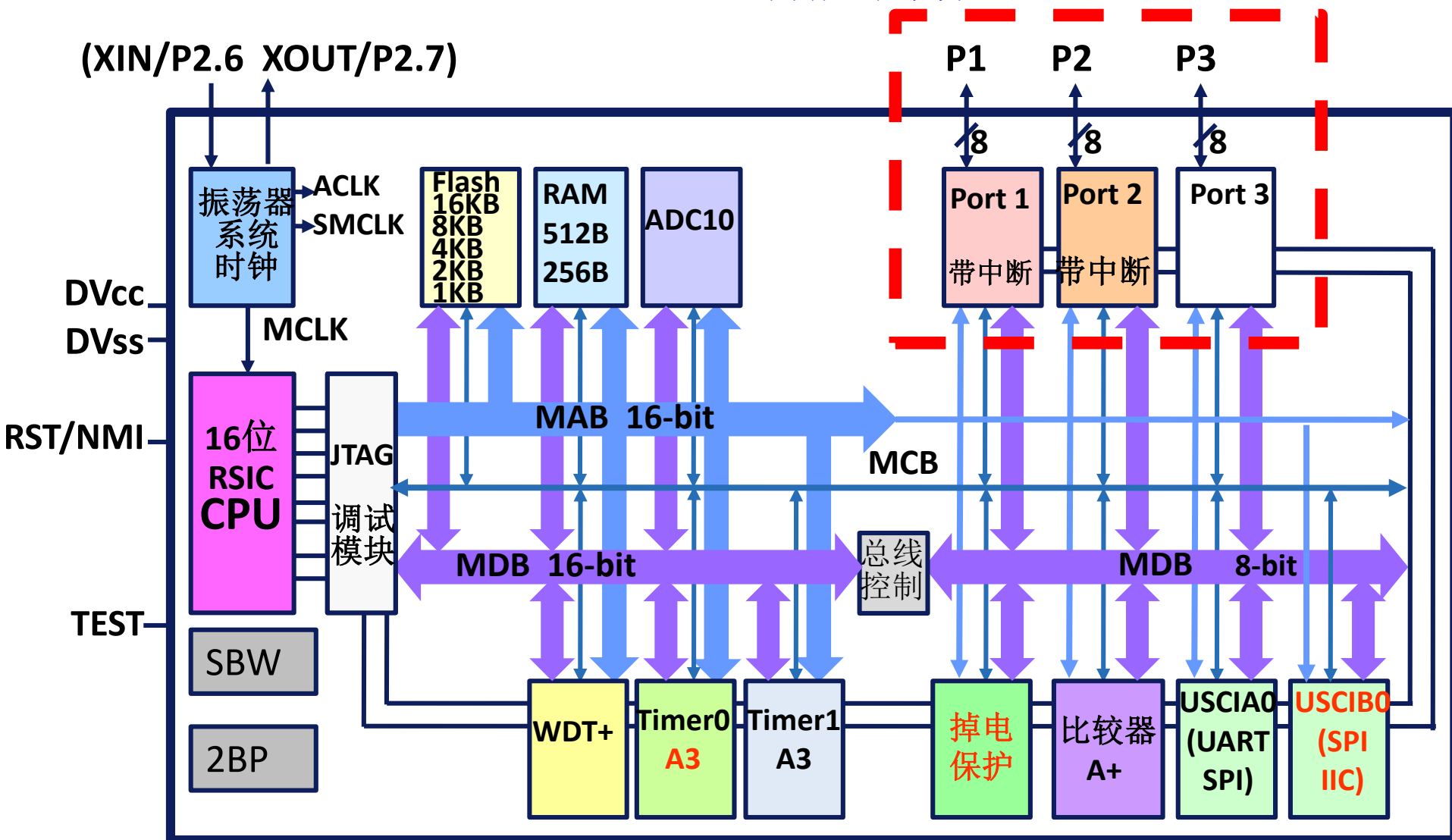


MSP430的基本输入/输出(I/O)

- MSP430有丰富的端口资源，
各产品因型号不同，所含端口资源不同,最多可达8个8位端口
MSP430G2xx有3个端口P1~P3，其中20个引脚封装只有P1~P2
- 目前MSP430系列单片机的总线不对外开放，
I/O端口的引脚是MCU对外进行输入/输出的重要通道
- MSP430系统中操作I/O端口的方法：

读写端口寄存器

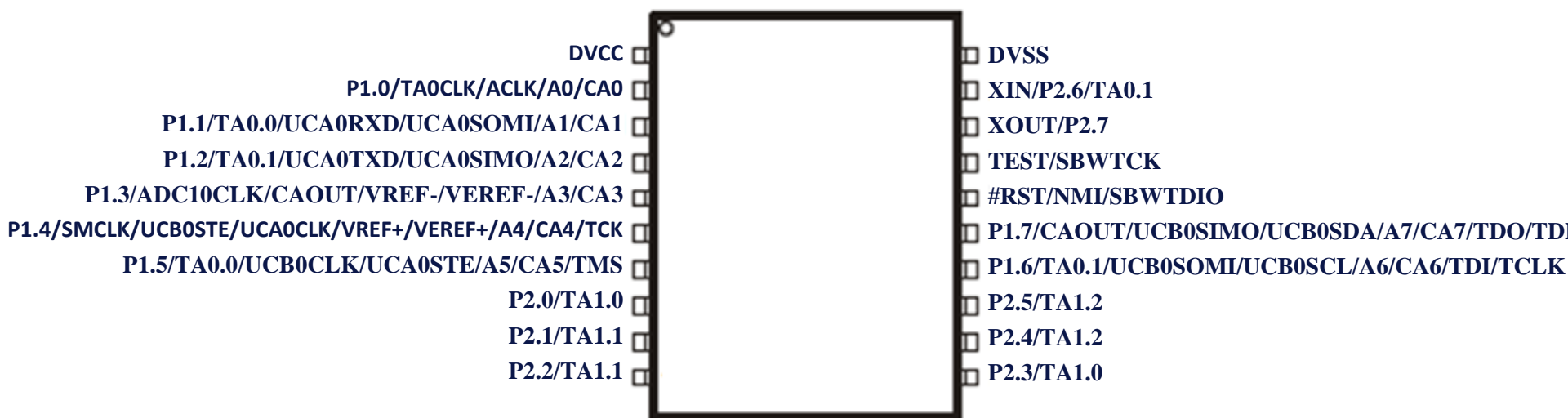
MSP430G2x53功能结构图



仅28引脚、32引脚的MSP430G2x53有端口3模块
20引脚的MSP430G2x53无端口3模块

MSP430G2553有3个I/O端口，每个端口对外有8根引脚
其中20引脚的MSP430G2553只有2个端口，28和32引脚的有3个端口

20引脚双列直插式 MSP430G2553



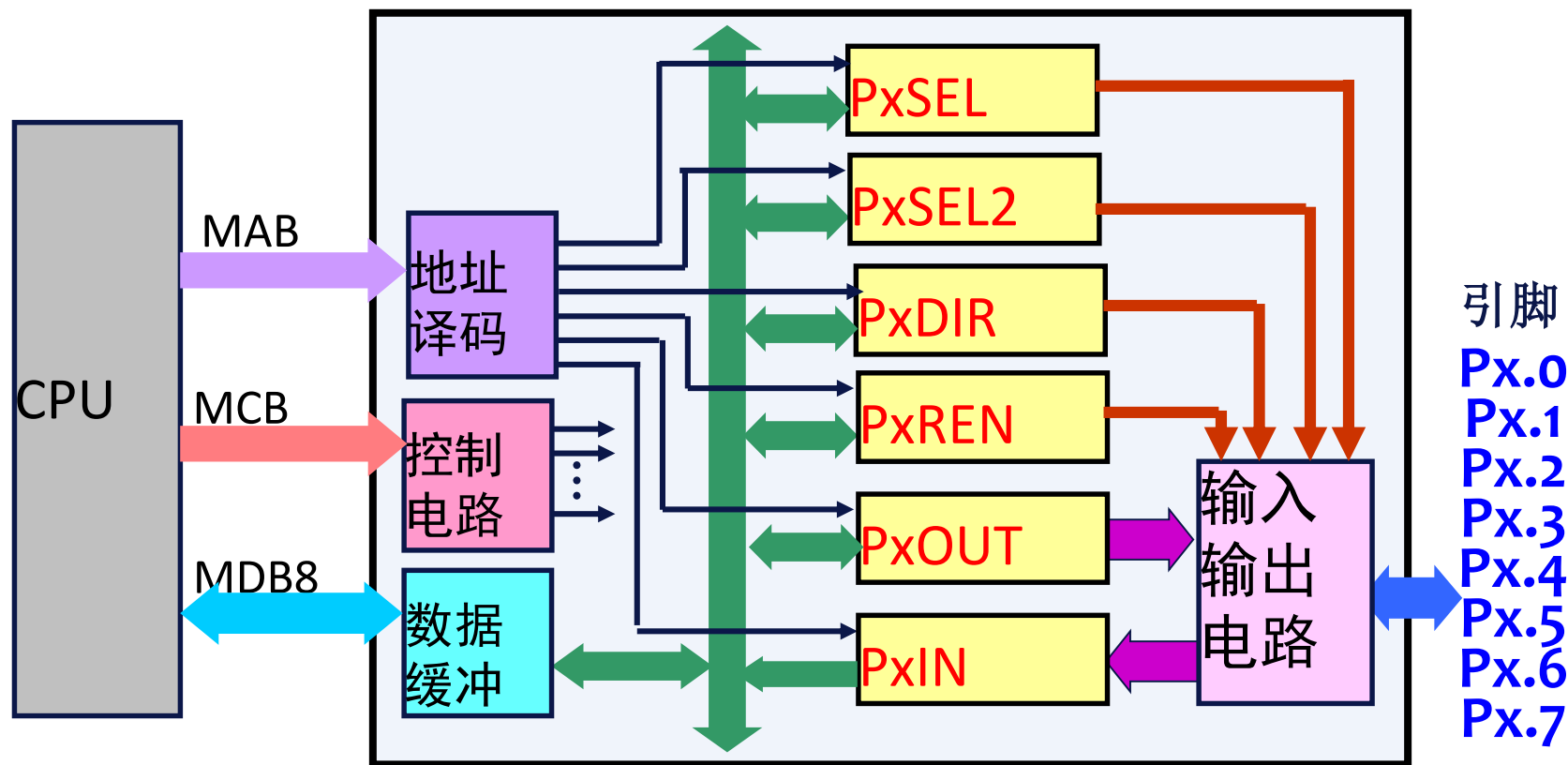
除去DVCC、DVSS、TEST、#RST这4个引脚外，
其余16个引脚均可作为基本输入/输出引脚。
参看 MSP430G2553.pdf P42~57

1. 管脚复用
2. 输入/输出复用

引脚号	MSP430G2553引脚功能
1	DVCC
2	P1.0/TA0CLK/ACLK/A0/CA0
3	P1.1/TA0.0/UCA0RXD/UCA0SOMI/A1/CA1
4	P1.2/TA0.1/UCA0TXD/UCA0SIMO/A2/CA2
5	P1.3/ADC10CLK/CAOUT/VREF-/VEREF-/A3/CA3
6	P1.4/SMCLK/UCB0STE/UCA0CLK/VREF+/VEREF+/A4/CA4/TCK
7	P1.5/TA0.0/UCB0CLK/UCA0STE/A5/CA5/TMS
8	P2.0/TA1.0
9	P2.1/TA1.1
10	P2.2/TA1.1
11	P2.3/TA1.0
12	P2.4/TA1.2
13	P2.5/TA1.2
14	P1.6/TA0.1/UCB0SOMI/UCB0SCL/A6/CA6/TDI/TCLK
15	P1.7/CAOUT/UCB0SIMO/UCB0SDA/A7/CA7/TDO/TDI
16	RST/NMI/SBWTDIO
17	TEST/SBWTCK
18	P2.7/XOUT
19	P2.6/XIN/TA0.1
20	DVSS

I/O端口x 基本输入/输出示意图(不考虑其他模块功能时)

General-purpose digital I/O (简称GPIO)



➤ **PxSEL**、**PxSEL2**功能选择寄存器：全为0：I/O功能；
其他：外设模块功能

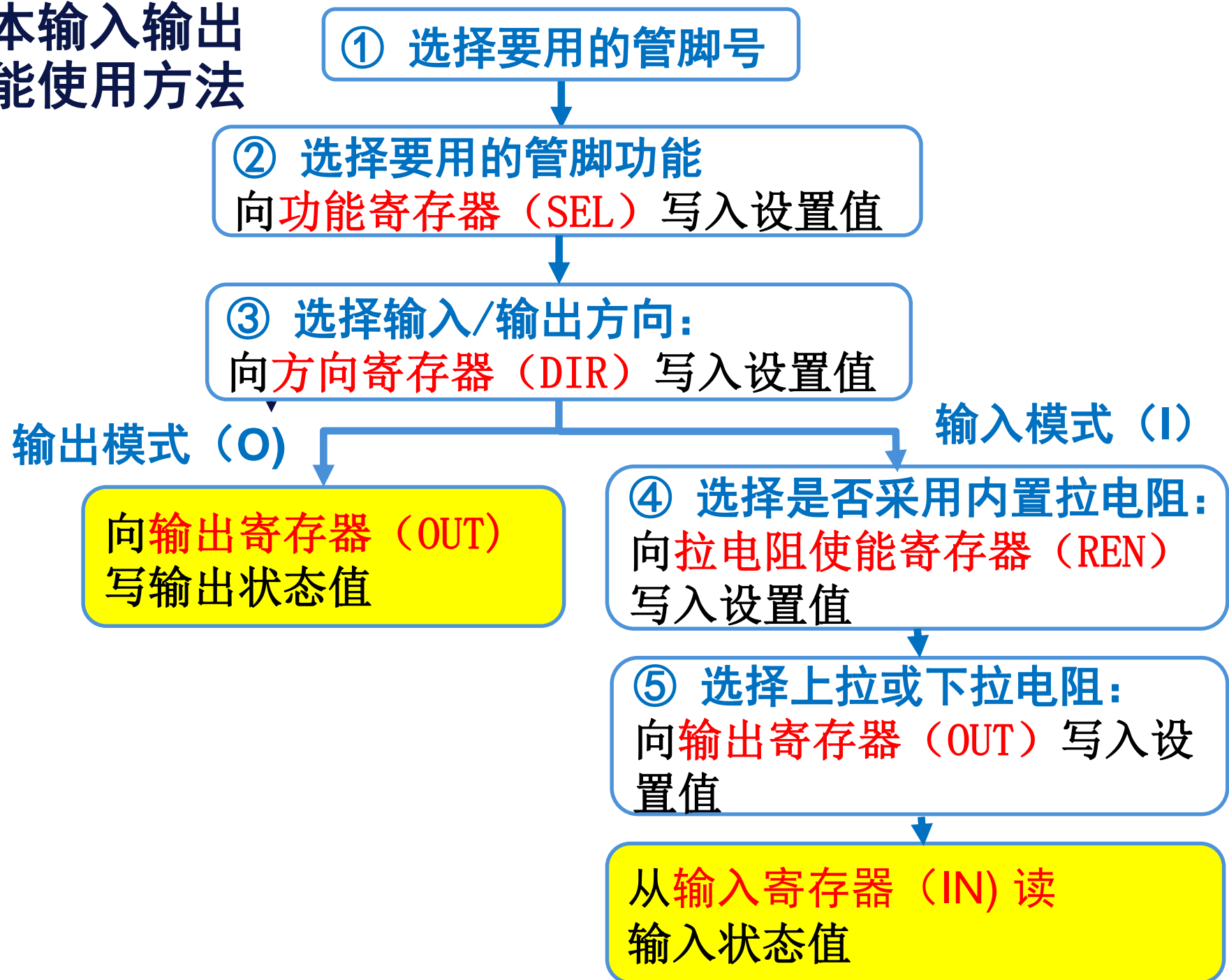
➤ **PxDIR**方向选择寄存器： 0: 输入； 1: 输出

➤ **PxIN** 输入寄存器： 0: 输入低电平； 1: 输入高电平

➤ **PxOUT**输出寄存器： 0: 输出低电平； 1: 输出高电平

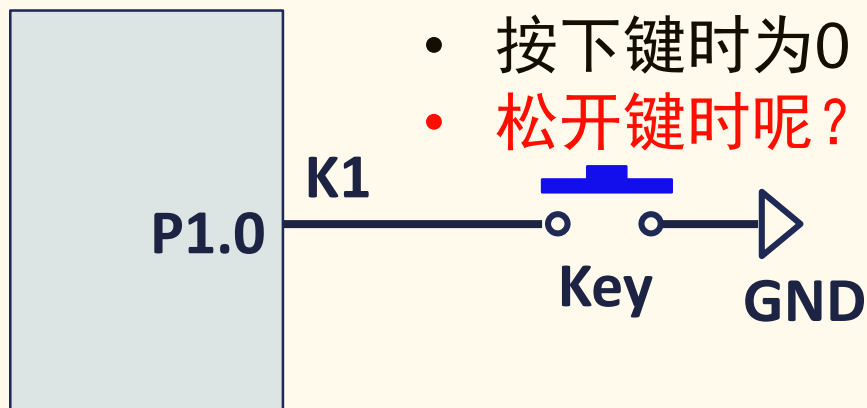
➤ **PxREN**上拉/下拉电阻使能寄存器： 0: 禁止； 1: 使能

基本输入输出 功能使用方法

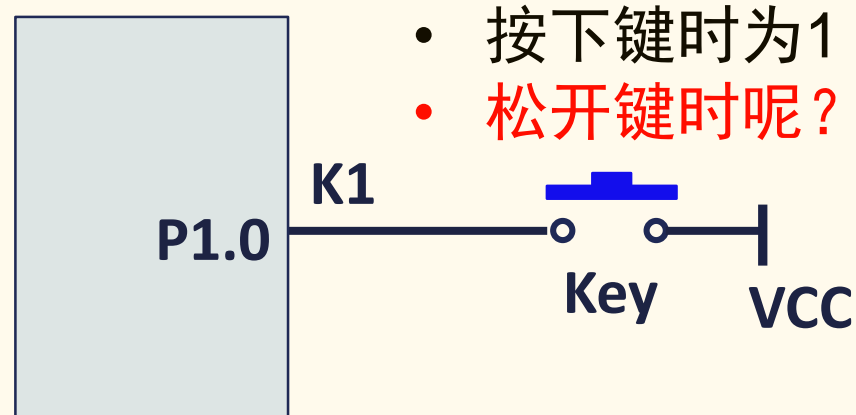


为什么输入功能时要更多的寄存器？

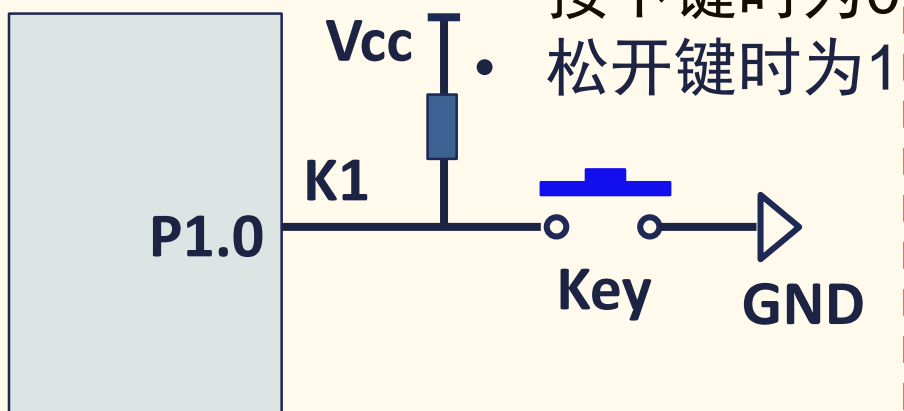
MSP430G2553



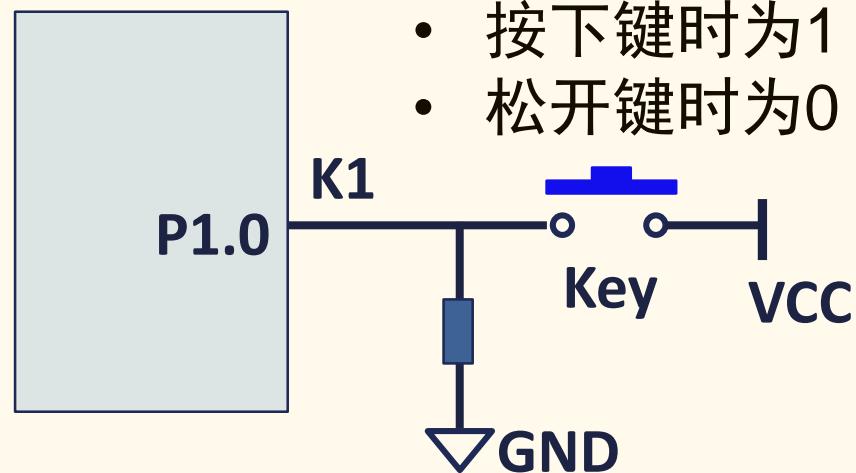
MSP430G2553



MSP430G2553



MSP430G2553

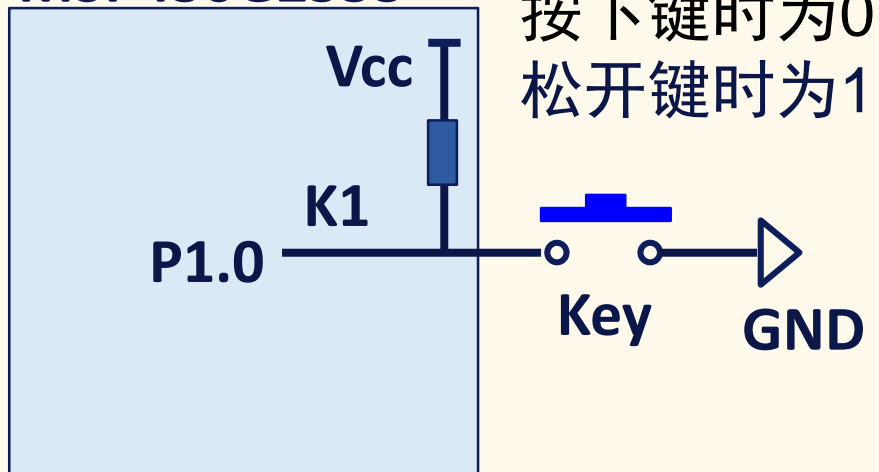


上拉电阻

下拉电阻

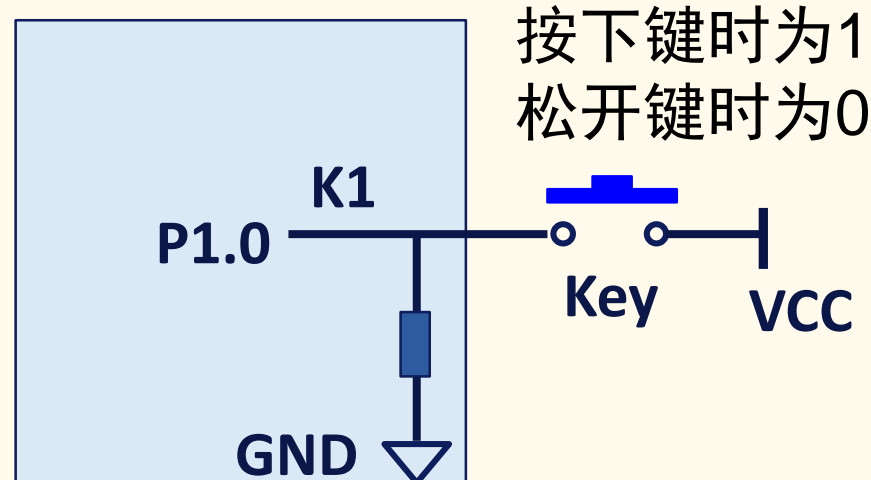
MSP430内部上拉或下拉电阻

MSP430G2553



内部上拉电阻

MSP430G2553



内部下拉电阻

- MSP430可以选择是否采用内部拉电阻（PxREN寄存器）
- 根据外部电路方式选择上拉或者下拉电阻（PxOUT寄存器）

基本输出功能使用方法（以P1端口为例）

① 选择要用的管脚号

P1.7 P1.6 P1.5 P1.4 P1.3 P1.2 P1.1 P1.0

② 选择要用的管脚功能（基本I/O功能）

P1SEL	7	6	5	4	3	2	1	0	管脚功能 选择 基本I/O时 都设置为0
	*	*	*	*	*	*	*	*	
P1SEL2	7	6	5	4	3	2	1	0	
	*	*	*	*	*	*	*	*	

③ 选择输入/输出方向（输出方向）

P1DIR	7	6	5	4	3	2	1	0	方向选择 输出方向：1
	*	*	*	*	*	*	*	*	

最后：写入输出状态

P1OUT	7	6	5	4	3	2	1	0	输出状态 根据需要 设为1或0
	*	*	*	*	*	*	*	*	

基本输出功能使用方法（以P1端口为例）演示：P1.3管脚按键

① 选择要用的管脚号

P1.7 P1.6 P1.5 P1.4 P1.3 P1.2 P1.1 P1.0

② 选择要用的管脚功能（基本I/O功能）

P1SEL

P1SEL2

7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	*
7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	*

管脚功能选择
基本IO时都设置为0

③ 选择输入/输出方向（输出方向）

P1DIR

7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	*

方向选择
输入方向：0

④ 选择是否采用内置拉电阻

P1REN

7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	*

拉电阻使能
采用：1
不采用：0

⑤ 选择上拉或者下拉电阻模式

P1OUT

7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	*

拉电阻模式
上拉：1
下拉：0

最终读入输入状态

P1IN

7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	*

输入状态1
或0

与I/O引脚基本输入/输出功能有关的I/O寄存器

- **PxSEL**功能选择寄存器 : P1SEL, P2SEL, P3SEL
- **PxSEL**功能选择寄存器2: P1SEL2, P2SEL2, P3SEL2
- **PxDIR**方向选择寄存器 : P1DIR, P2DIR, P3DIR
- **PxOUT**输出数据寄存器 : P1OUT, P2OUT, P3OUT
- **PxIN**输入数据寄存器 : P1IN, P2IN, P3IN
- **PxREN**上下拉电阻使能 : P1REN, P2REN, P3REN

▲x可为1~3, 由具体的单片机确定

对于P1端口, 有P1SEL, P1SEL2, P1DIR, P1OUT, P1IN, P1REN

▲均为8位寄存器

▲按位配置: 端口的每根引脚可单独配置, 相互之间不影响

■ PxSEL功能选择寄存器

■ PxSEL2功能选择寄存器2

为减少MCU的对外引脚，
端口P1~P3的引脚有多种功能，即被MCU内的多个模块复用，
两个功能选择寄存器就是用于选择引脚的功能

位	7	6	5	4	3	2	1	0
	PxSEL.7	PxSEL.6	PxSEL.5	PxSEL.4	PxSEL.3	PxSEL.2	PxSEL.1	PxSEL.0
初始值	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

位	7	6	5	4	3	2	1	0
	PxSEL2.7	PxSEL2.6	PxSEL2.5	PxSEL2.4	PxSEL2.3	PxSEL2.2	PxSEL2.1	PxSEL2.0
初始值	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

■ PxDIR 方向选择寄存器

P1~P3端口为单向端口，
通过方向寄存器**选择端口引脚的输入/输出方向**

位	7	6	5	4	3	2	1	0
	PxDIR.7	PxDIR.6	PxDIR.5	PxDIR.4	PxDIR.3	PxDIR.2	PxDIR.1	PxDIR.0
复位值	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

PxDIR.y = 0, 端口**x**的引脚**y**选择为**输入方向**

PxDIR.y = 1, 端口**x**的引脚**y**选择为**输出方向**

例 如果**P1DIR**的内容为**00h**, 表示端口P1的**8个引脚**均用于**输入**
如果**P2DIR**的内容为**FFh**, 表示端口P2的**8个引脚**均用于**输出**
如果**P3DIR**的内容为**0Fh**, 表示端口P3的**P3.7~P3.4**用于**输入**
端口P3的**P3.3~P3.0**用于**输出**

■ PxOUT输出寄存器：决定输出到引脚信号上的电平

- 是端口的输出缓冲寄存器，
- 当端口某位的DIR设置为输出时，
该位端口引脚的状态对应为输出寄存器相应位的值。
- 编程改变输出寄存器该位的值就可改变该位引脚的输出状态

位	7	6	5	4	3	2	1	0
	PxOUT.7	PxOUT.6	PxOUT.5	PxOUT.4	PxOUT.3	PxOUT.2	PxOUT.1	PxOUT.0
复位值	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

PxOUT.y = 0 , 端口x的引脚y输出低

PxOUT.y = 1 , 端口x的引脚y输出高

例 如果P2OUT的内容为00h,
表示向端口P2的8个引脚均输出0, 即低电平

如果P2OUT的内容为0Fh,
表示向端口P2的引脚P2.7~P2.4输出0, 即低电平
表示向端口P2的引脚P2.3~P2.0输出1, 即高电平

■ PxIN输入寄存器: 获取引脚信号上输入的电平

- 是端口的输入缓冲寄存器;
- 当端口某位的DIR设置为输入时,
输入寄存器相应位的值对应为该引脚的输入状态;
- 通过编程读取该寄存器对应位的值, 就可获得相应引脚的状态

位	7	6	5	4	3	2	1	0
r	PxIN.7	PxIN.6	PxIN.5	PxIN.4	PxIN.3	PxIN.2	PxIN.1	PxIN.0

$PxIN.y = 0$, 端口x的引脚y输入为低

$PxIN.y = 1$, 端口x的引脚y输入为高

例 如果P1IN的内容为00h,

表示端口P1的8个引脚的状态均为0, 即低电平

如果P1IN的内容为0Fh,

表示端口P1的引脚P1.7~P1.4的状态为0, 即低电平

端口P1的引脚P1.3~P1.0的状态为1, 即高电平

PxREN上拉或下拉电阻使能寄存器

▲使能或禁止相应I/O引脚MCU内的上拉或下拉电阻;

位	7	6	5	4	3	2	1	0
	PxREN.7	PxREN.6	PxREN.5	PxREN.4	PxREN.3	PxREN.2	PxREN.1	PxREN.0
复位值	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

PxREN.y = 1,使能端口**x**的引脚**y**上拉或下拉电阻;

PxREN.y = 0,禁止端口**x**的引脚**y**上拉或下拉电阻;

▲使能引脚上拉或下拉功能后,

通过设置**PxOUT.y**相应位来选择上拉或下拉

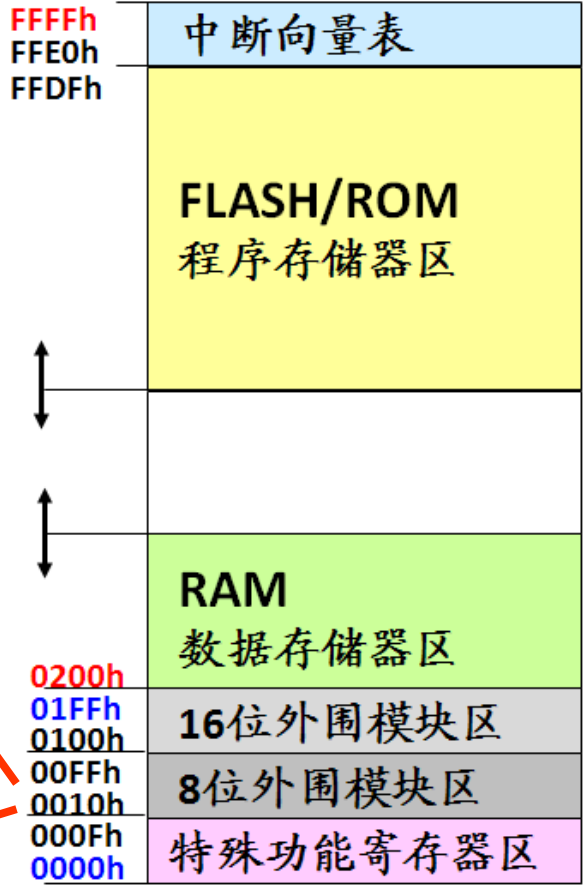
PxOUT.y = 1,选择端口**x**的引脚**y**通过电阻接到Vcc电源上, 把电平拉高;

PxOUT.y = 0,选择端口**x**的引脚**y**通过电阻接到Vss地线上, 把电平拉低;

与基本I/O功能有关的端口属性、复位值和地址

- PxSEL, PxDIR, PxOUT, PxREN, PxIN等只是起的名字
- 每个寄存器有一个地址编号

端口	SEL 功能	DIR 方向	OUT 输出	IN 输入
属性	可读写	可读写	可读写	只读
复位值	00H	00H	-	-
P1	026H	022H	021H	020H
P2	02EH	02AH	029H	028H
P3	01BH	01AH	019H	018H

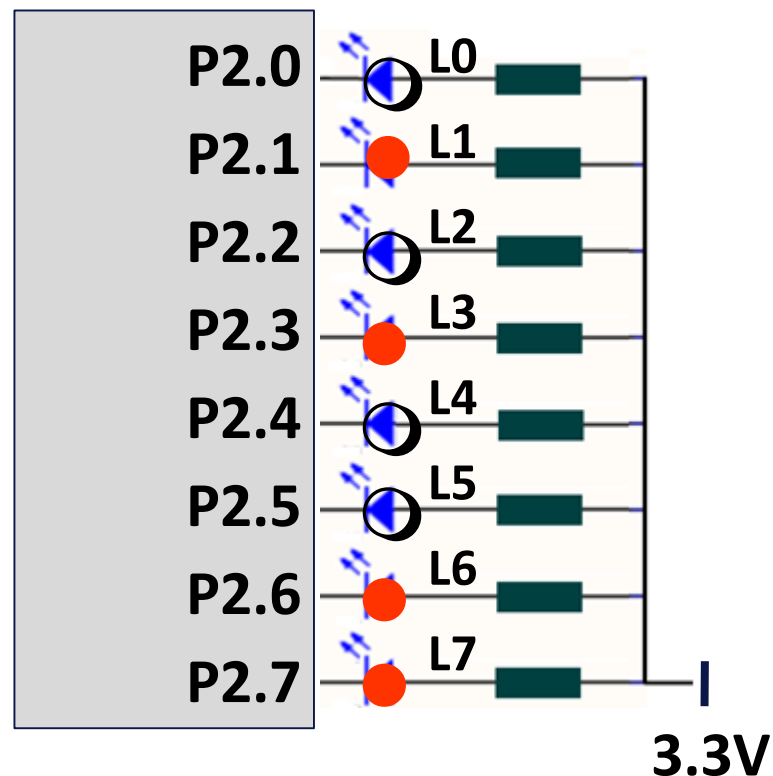


- 复位后, 各端口作为基本I/O输入端口
- 各I/O端口均为8位, 只能用字节操作指令

8个发光二极管控制电路

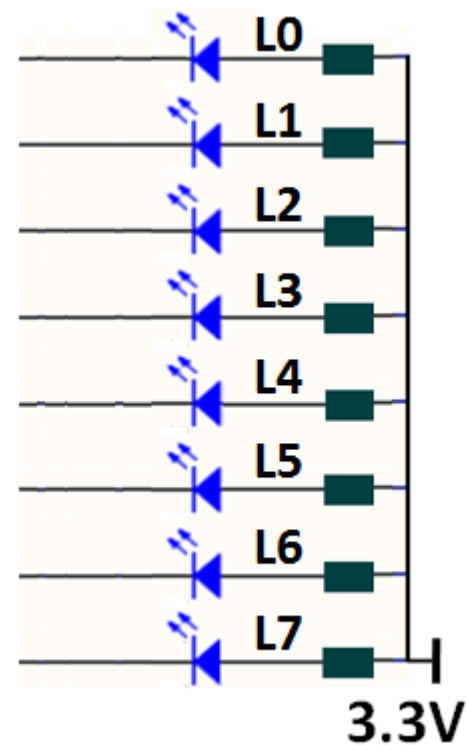
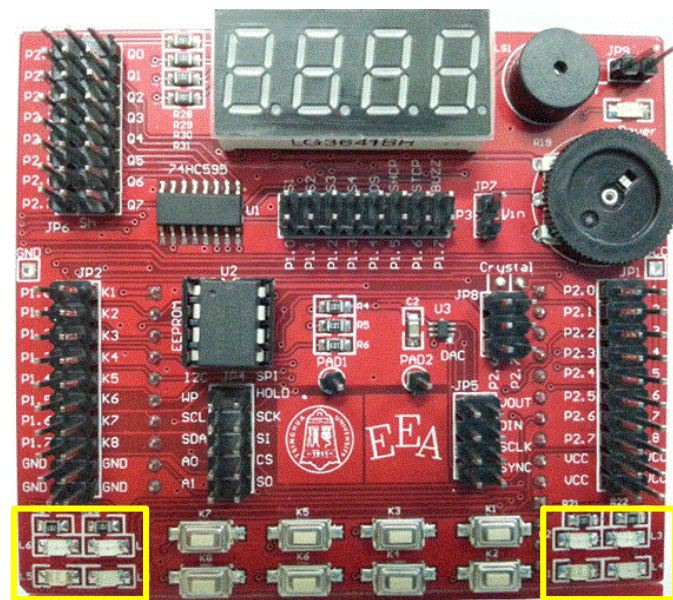
如果用
MSP430G2553单片机
引脚P2.7~P2.0分别与
这8个发光二极管相连

MSP430G2553



设置I/O寄存器P2SEL、P2SEL2、P2DIR的值，并改变P2OUT的值，
就能改变与二极管连接的引脚P2.7~P2.0的电压，
从而改变发光二极管的状态。

实验板上发光二极管电路

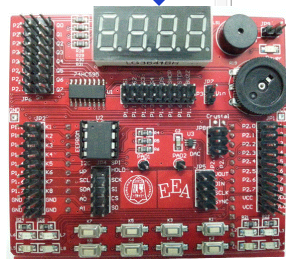


用导线将P2.7~P2.0分别与8个二极管连接



Code Composer Studio (CCS)

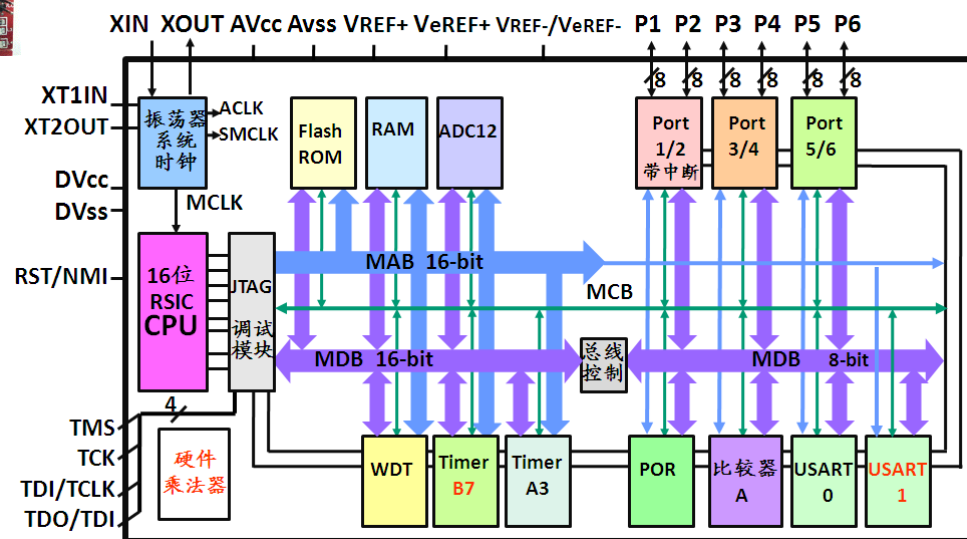
目标对象
(实验系统)



PC机



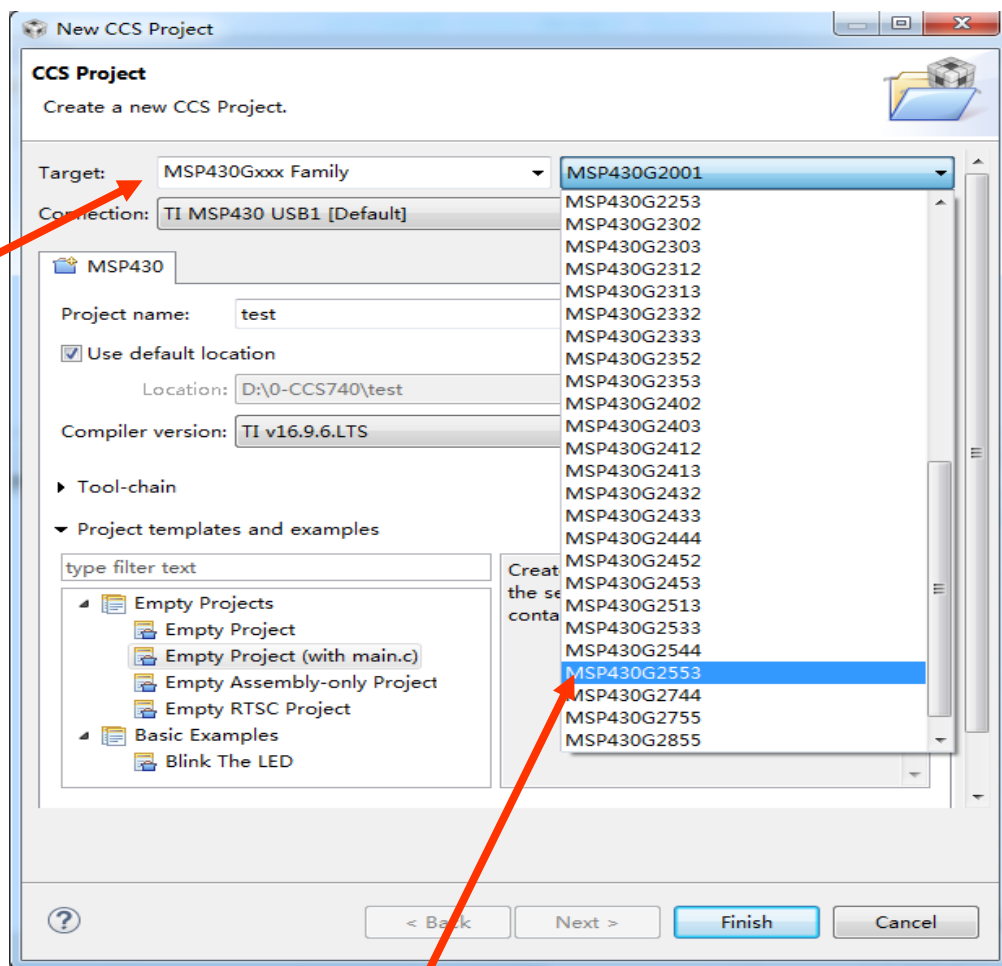
PC机侧利用CCS软件
在调试器的支持下实现
对单片机的编程和调试



单片机结构图

在CCS开发环境下，
对项目所用的单片机选择型号

选系列



选型号

除课件外，MSP430常用的学习资料

□**用户手册** Msp430xxxx family user's guide.pdf

包括该系列MCU内部各模块功能、操作及端口寄存器的详细介绍

□**单片机芯片数据手册** msp430xxxx.pdf 基本特性说明PDF文档

包括 引脚描述、功能模块、操作条件、电气特性等

□**Ti公司提供的示例程序**

□**网上相关的Msp430学习论坛**

□**相关的一些MSP430书籍**

MCU工程师炼成记,丁武峰,机械工业出版社, 2013

MSP430超低功耗16位单片机开发实例,唐继贤,北航出版社
,2014

谢谢!

Q & A

