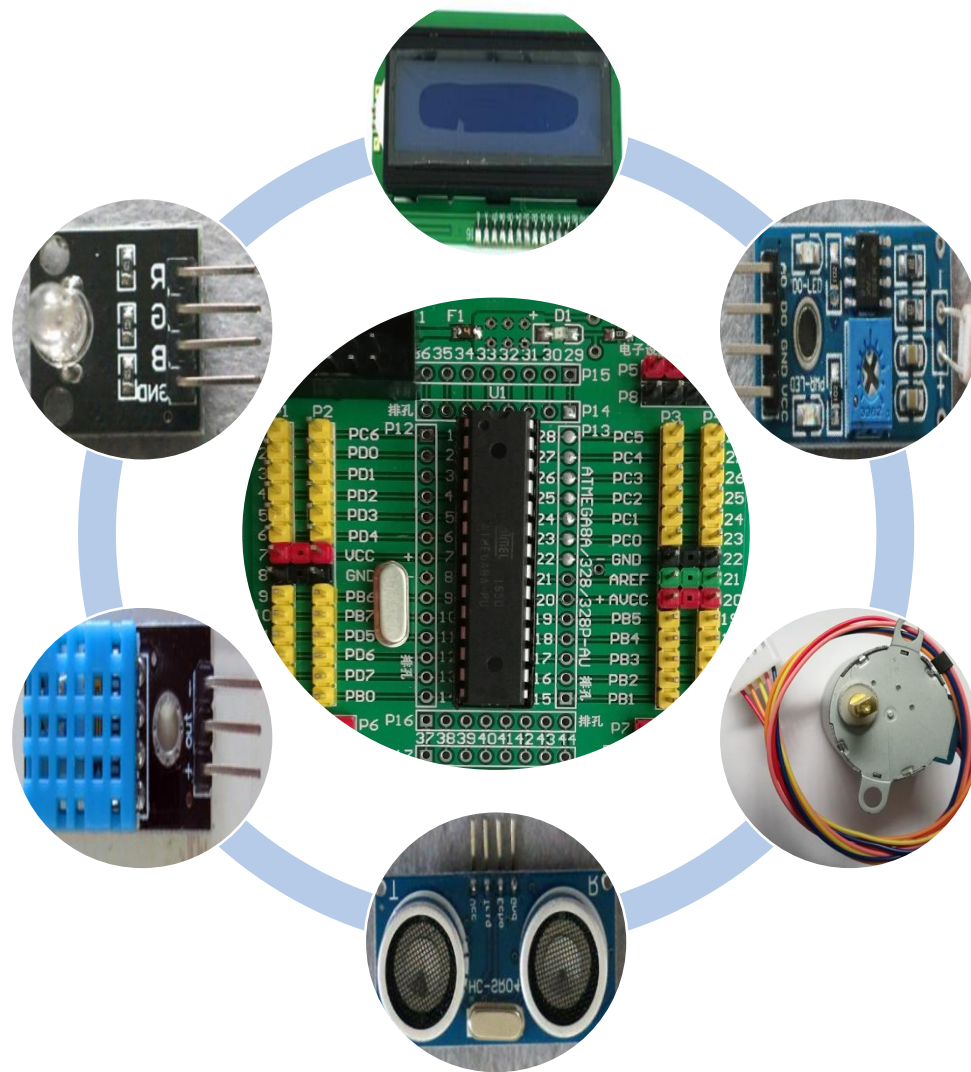


电子设计实践 基础

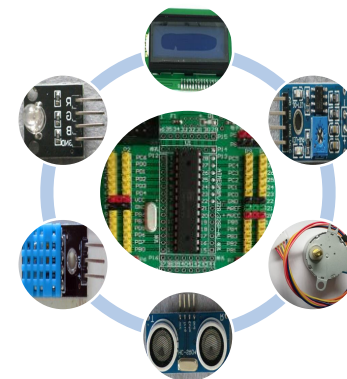
MCU与 RGB LED和触摸开关 之C语言编程



上节课内容回顾

■了解MCU芯片(ATmega8A)与其程序编写

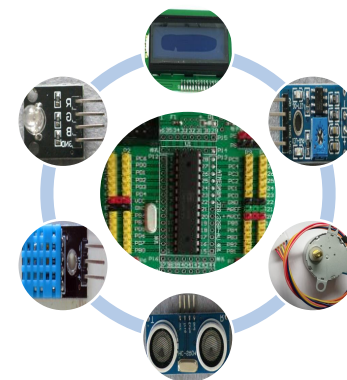
- ATmega8A的结构：8位MCU
- ATmega8A的寄存器：32, 64(IO)
- ATmega8A的指令：RISC
- ATmega8A内部存储器：8KB, 1KB, 512B
- ATmega8A的系统时钟:默认1M,2/4/8M;16M
- ATmega8A的IO端口与C语言编程
- ATmega8A的程序下载（烧写）



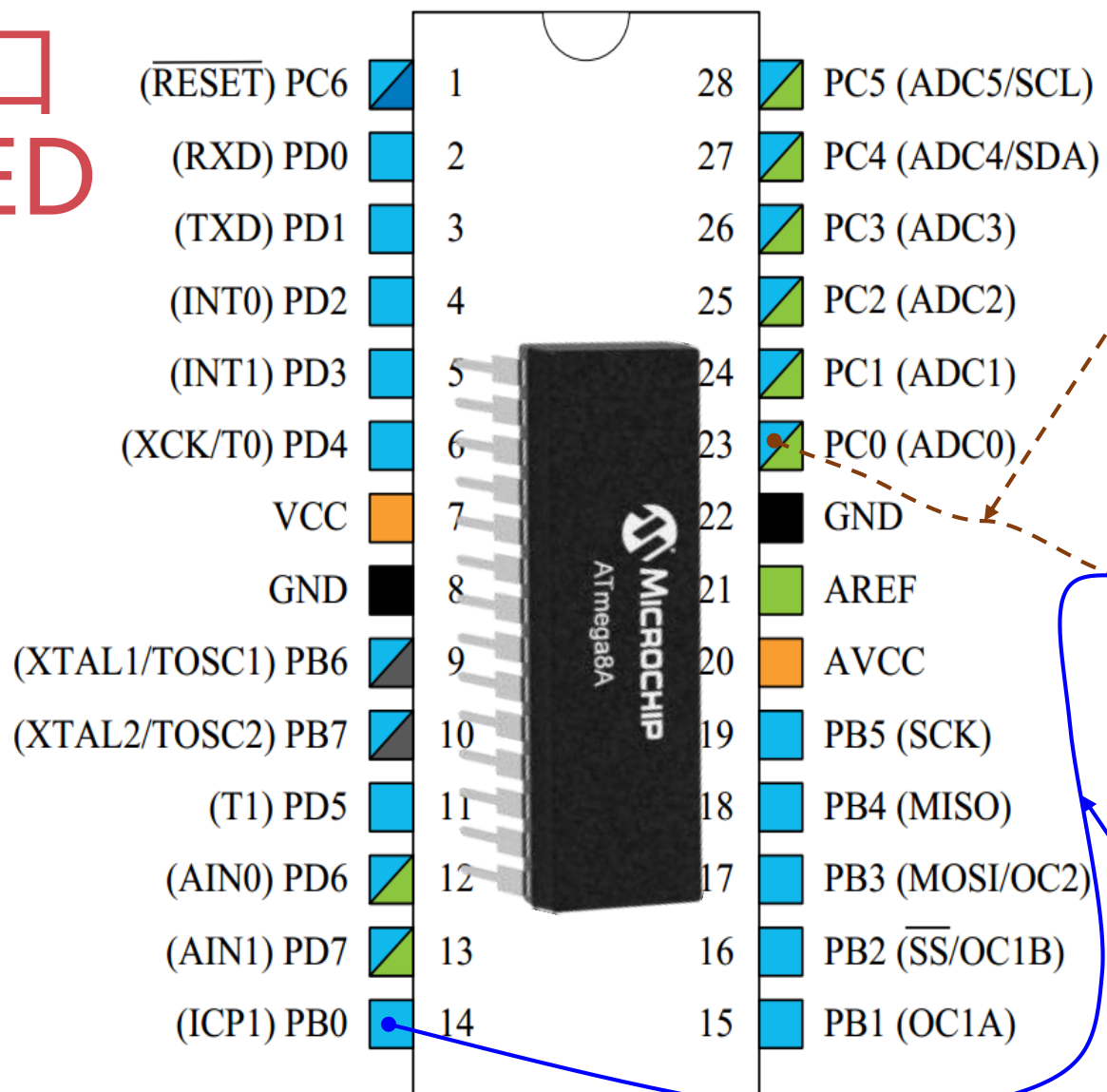
本节课主要内容

■ MCU芯片与RGB LED和触摸开关及其编程

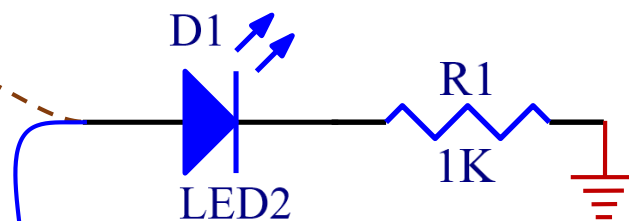
- `#include <avr/io.h>`
- IO端口控制与编程总结
- C语言的位运算
- RGB LED及其编程
- 一位触摸开关及其编程



IO端口 控制LED



寄存器名	偏移地址
PORTC	0x15
DDRC	0x14
PINC	0x13



寄存器名	偏移地址
PORTB	0x18
DDRB	0x17
PINB	0x16

IO端口控制LED

```
int main(void)
```

```
{
```

```
int i,j;
```

```
*(volatile unsigned char *)(0x20+0x14) = 0x01; //DDRC[0] = 1
```

冗长

管脚由PC0
换成PB0

DDRB:(0x20+0x17)

#define DDRC ...

```
while (1)
```

```
{
```

```
*(volatile unsigned char *)(0x20+0x15) = 0x01; //PORTC[0] = 1;
```

```
for(i=0;i<100;i++) for(j=0;j<1000;j++);
```

```
*(volatile unsigned char *)(0x20+0x15) = 0x00; //PORTC[0] = 0;
```

```
for(i=0;i<100;i++) for(j=0;j<1000;j++);
```

```
}
```

```
}
```

PORTB:(0x20+0x18)



#include <avr/io.h>

```
#include <avr/io.h>
```

```
int main(void)
```

```
{
```

```
#define DDRC *(volatile unsigned char *)(0x20+0x14)
.....
```

```
int i,j;
```

```
DDRC = 0x01; //DDRC[0] = 1
```

```
while (1)
```

```
{
```

```
PORTC = 0x01; //PORTC[0] = 1;
```

```
for(i=0;i<100;i++) for(j=0;j<1000;j++);
```

```
PORTC = 0x00; //PORTC[0] = 0;
```

```
for(i=0;i<100;i++) for(j=0;j<1000;j++);
```

```
}
```

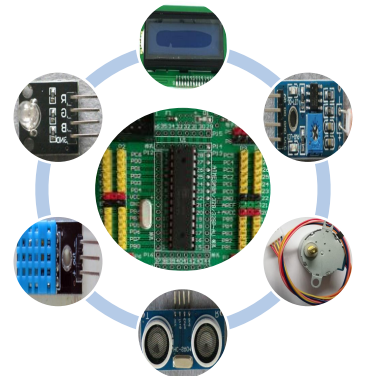
```
}
```

DDRB...

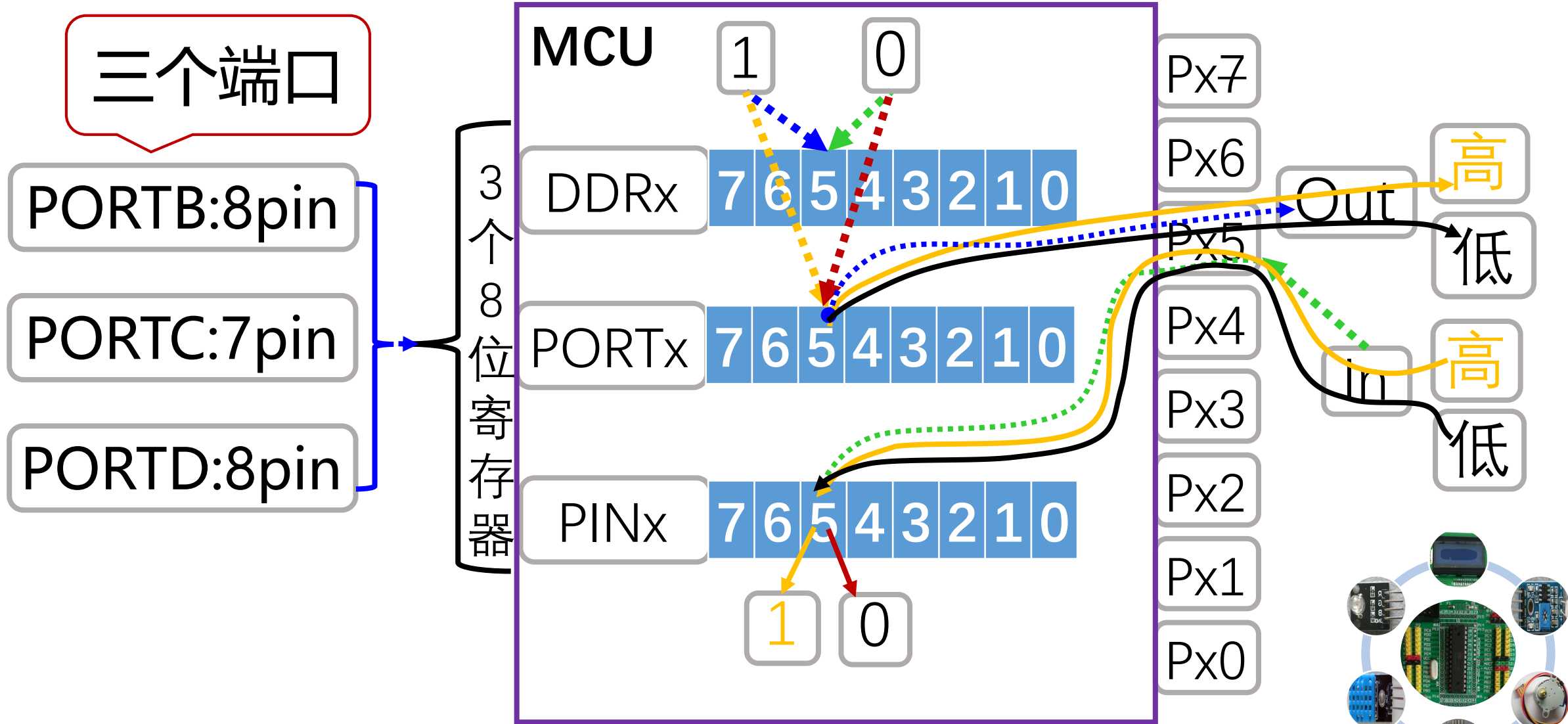
如管脚由PC0换成PB0?

PORTB...

如何用PD2管脚控制LED的亮灭?

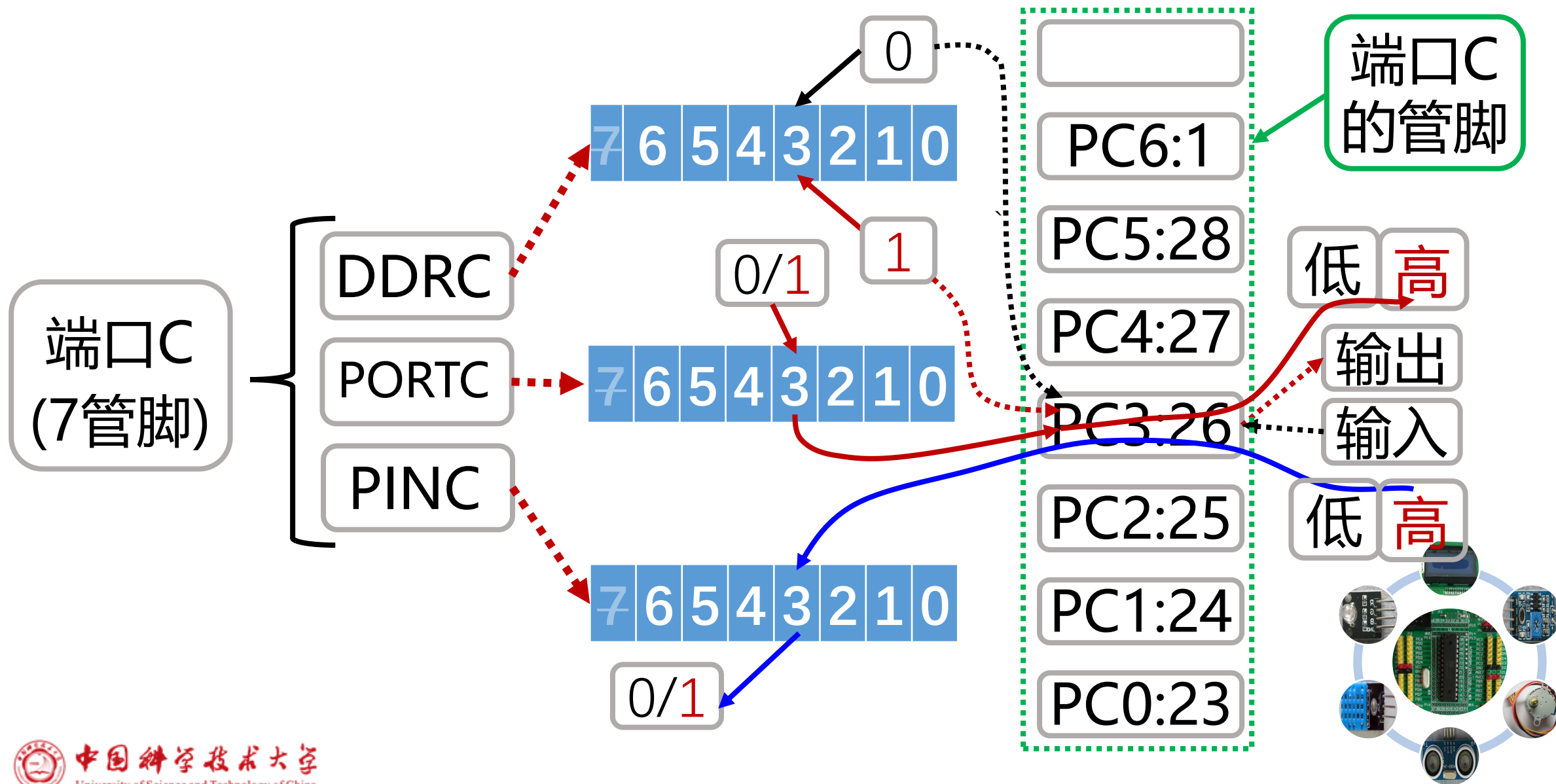


IO端口控制与编程总结

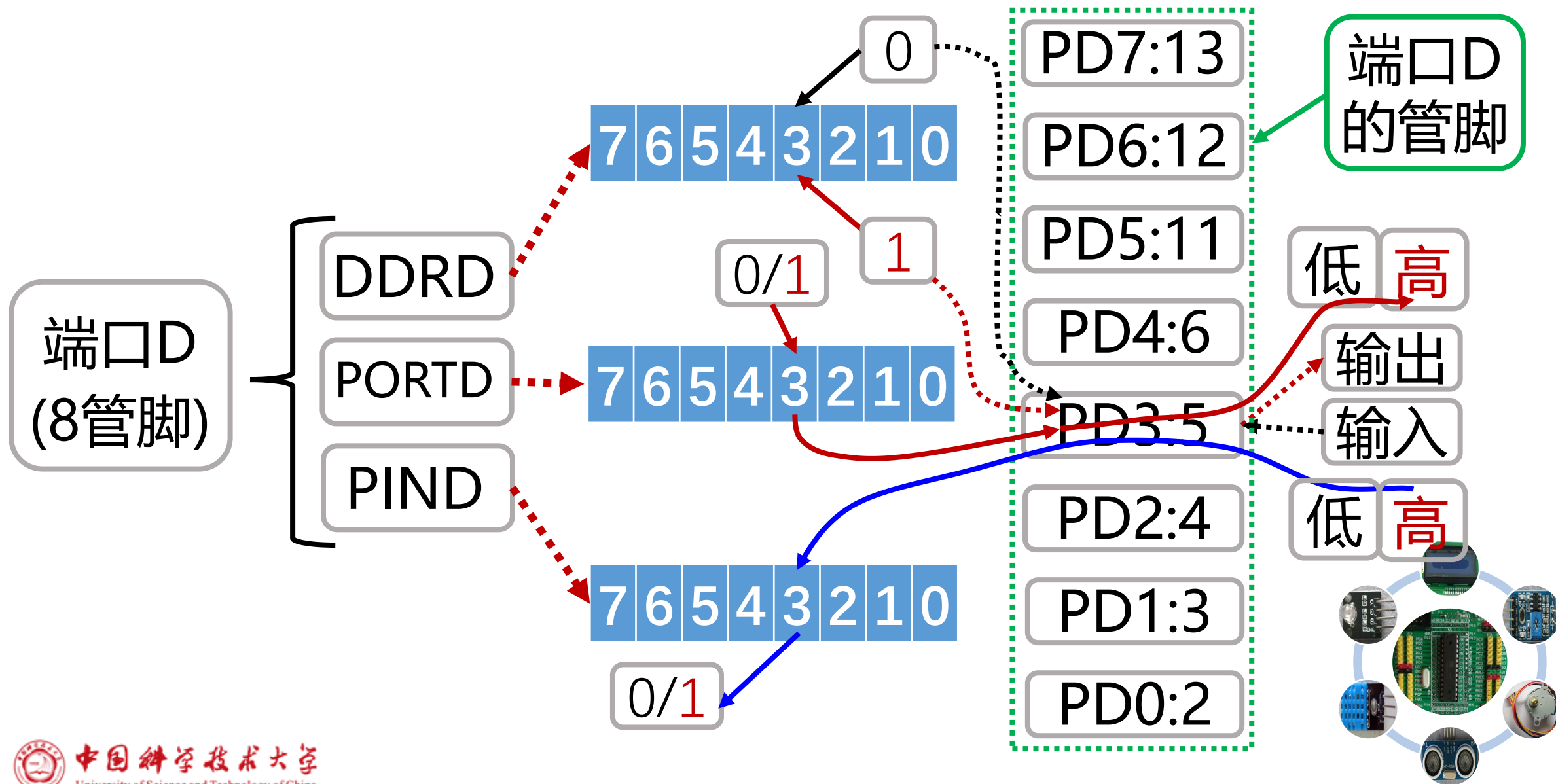


注：x为B、C或D

IO端口控制与编程总结：端口C



IO端口控制与编程总结：端口D



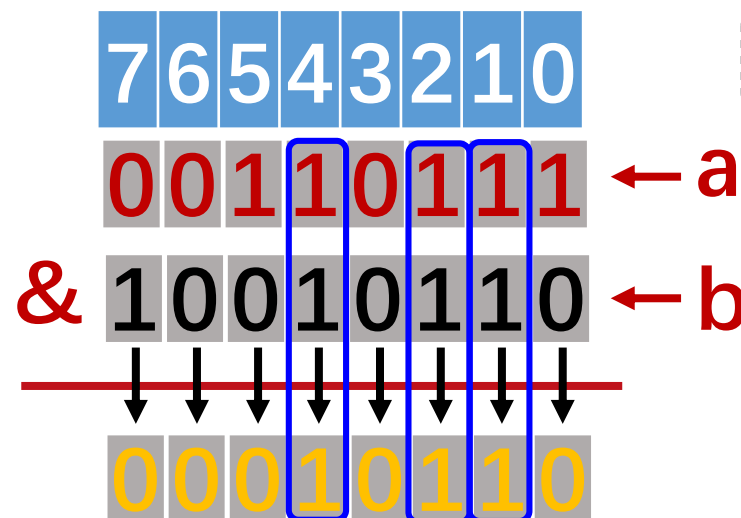
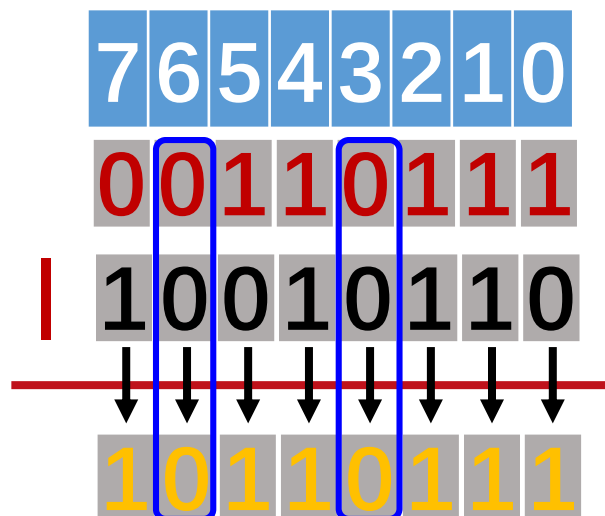
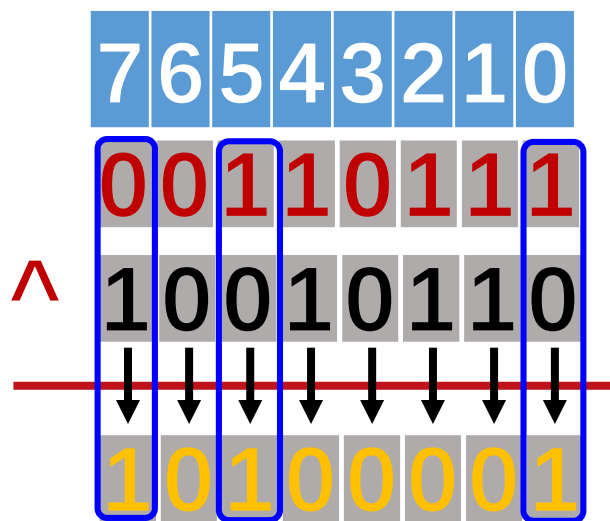
C语言的位运算

• 二进制按位运算

C语言位运算

&, |, ^, ~

<<, >>



unsigned char a,b;

a=55

b=150

10进制

a=00110111

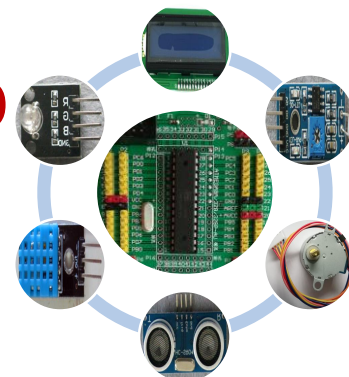
b=10010110

2进制

a=0x37

b=0x96

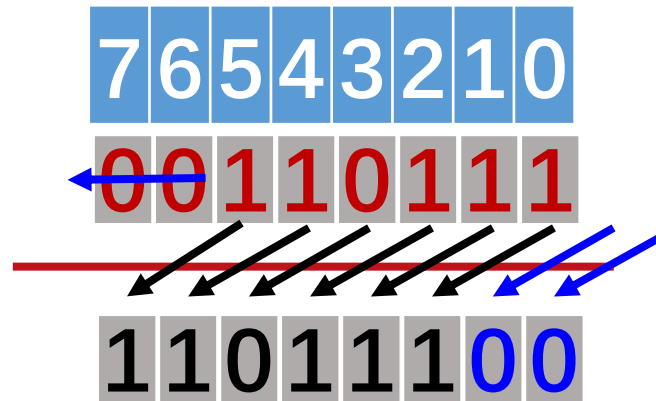
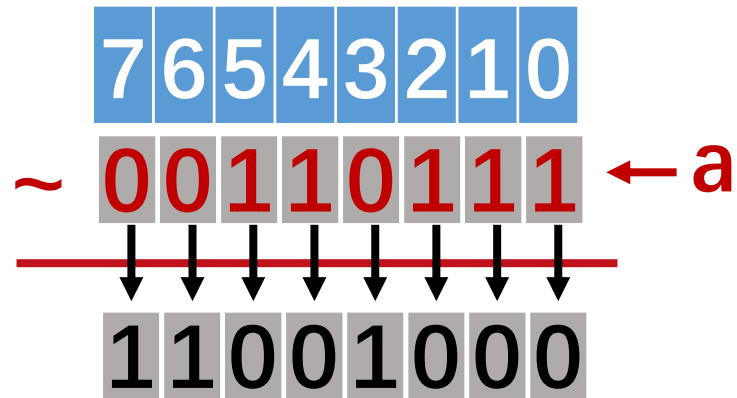
16进制



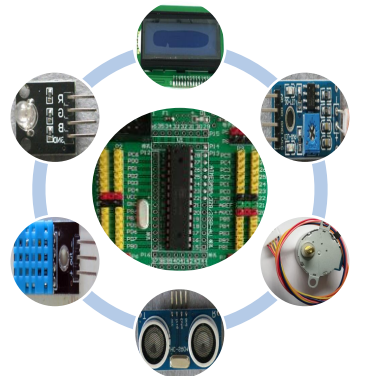
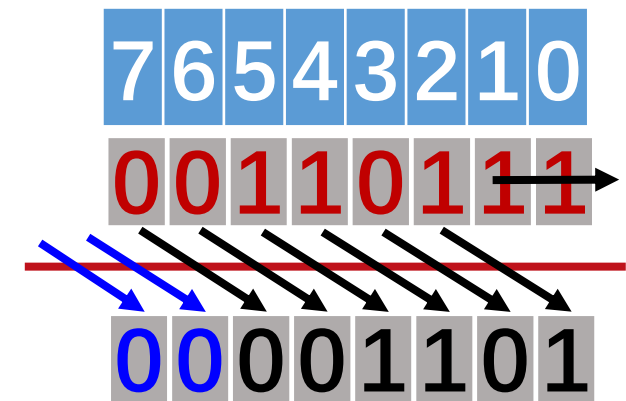
C语言的位运算

- 二进制按位计算

$a \ll 2$

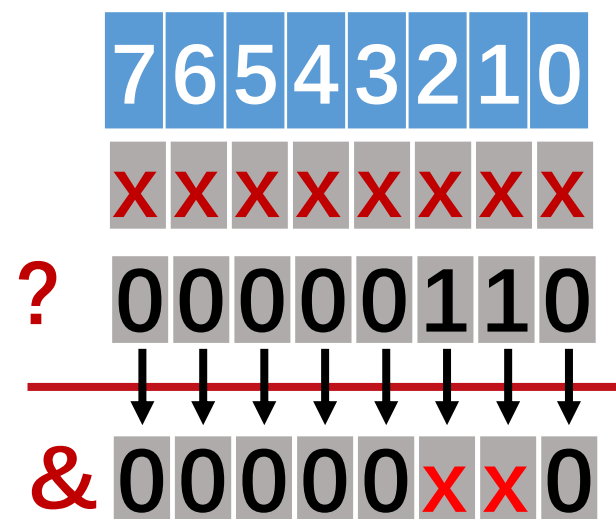
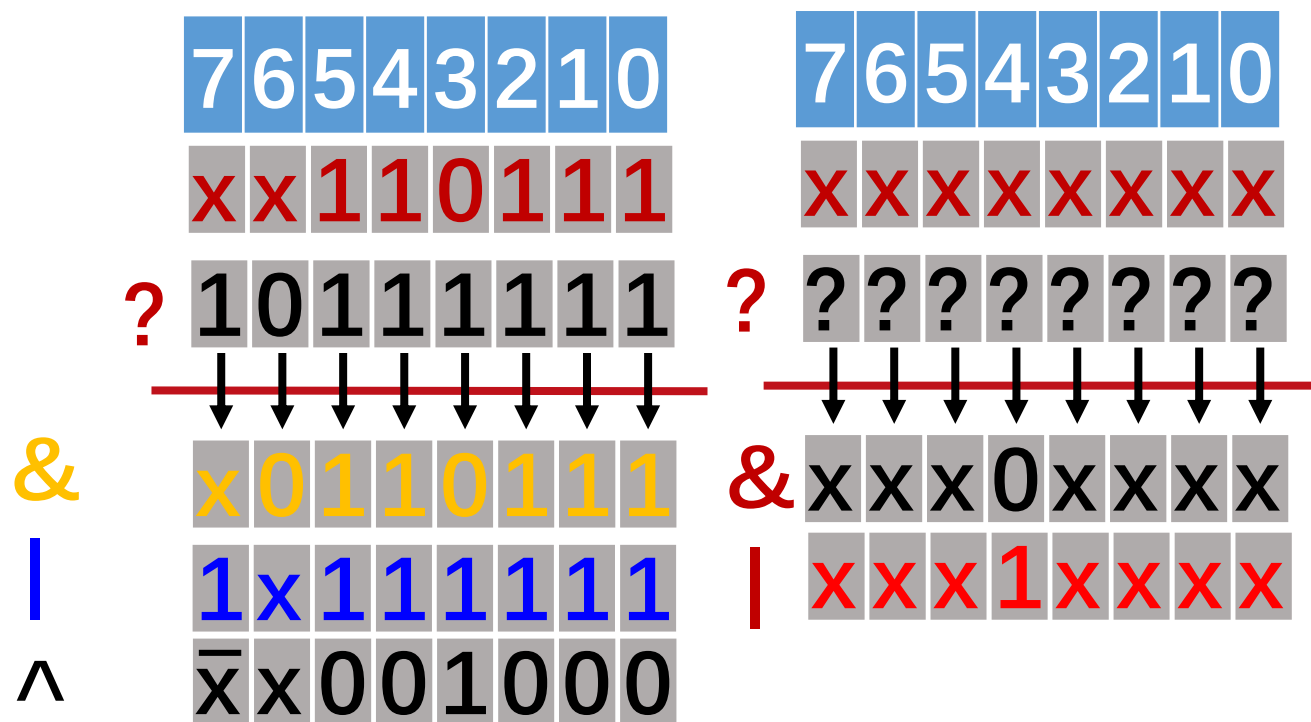


$a \gg 2$



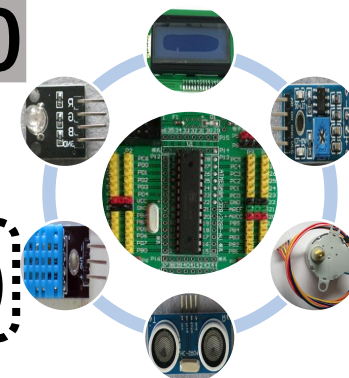
C语言位运算在单片机编程时的意义

- 改变指定位的值：清零/置位/取反
- 改变指定位的值，保护其它位
- 取指定位的值

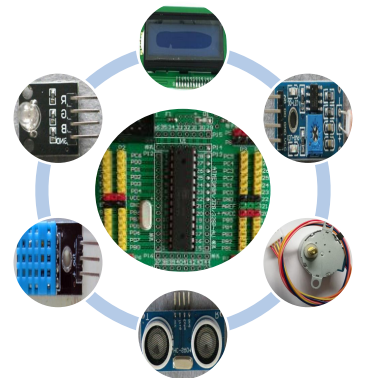
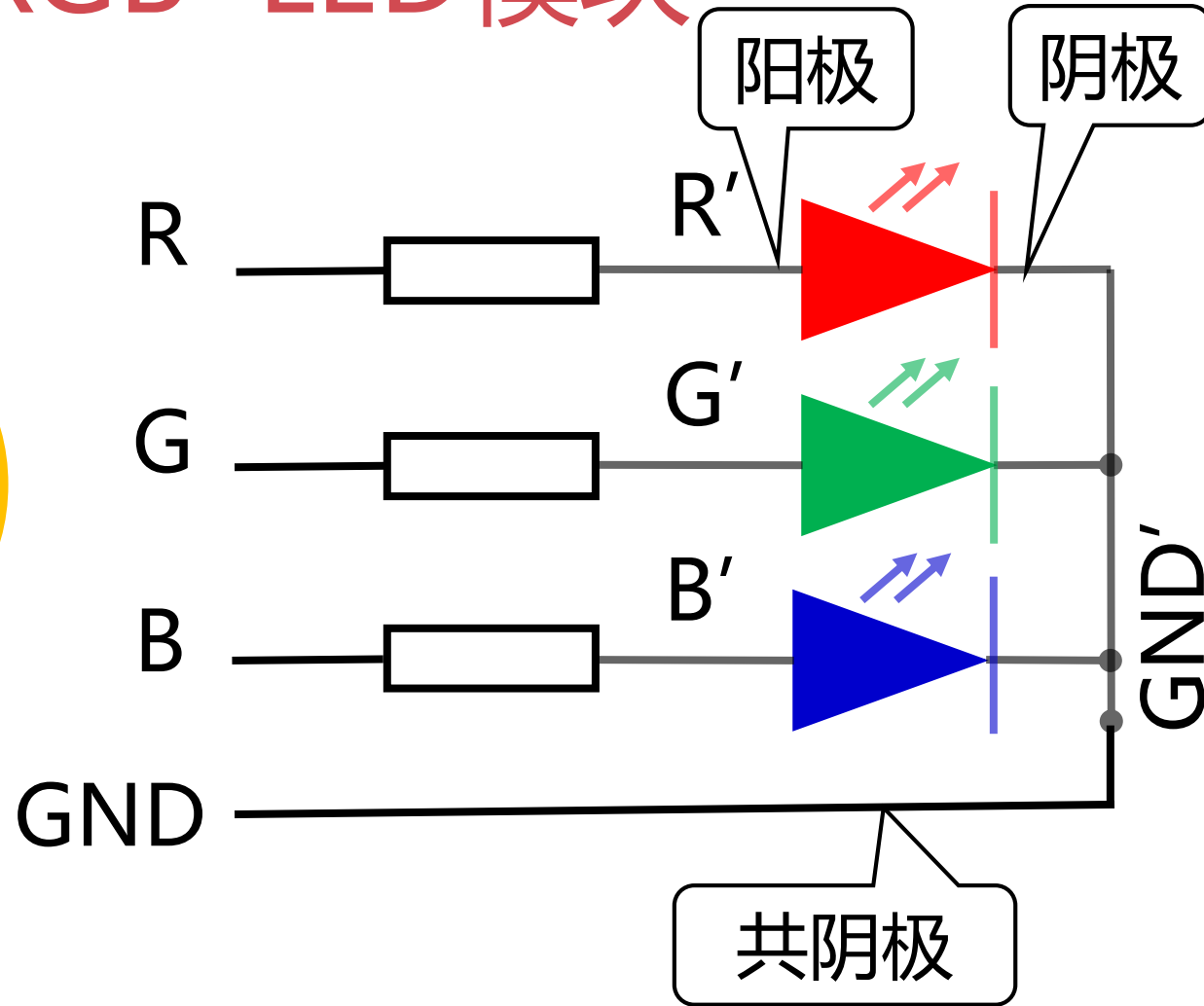
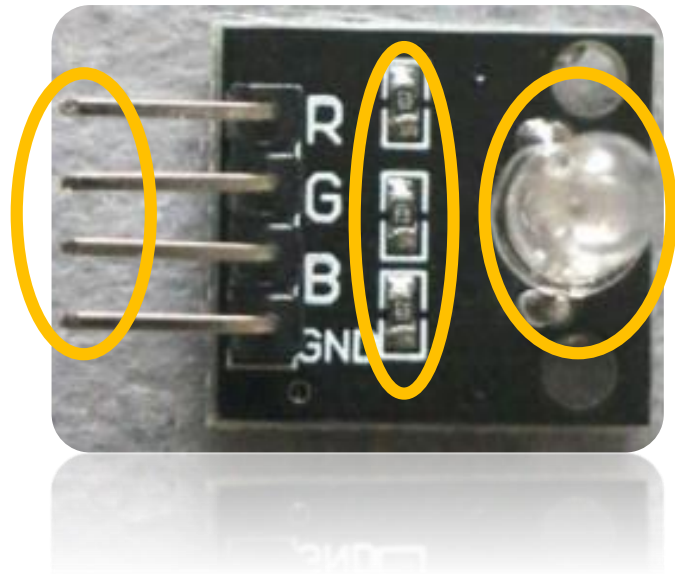


• 定位

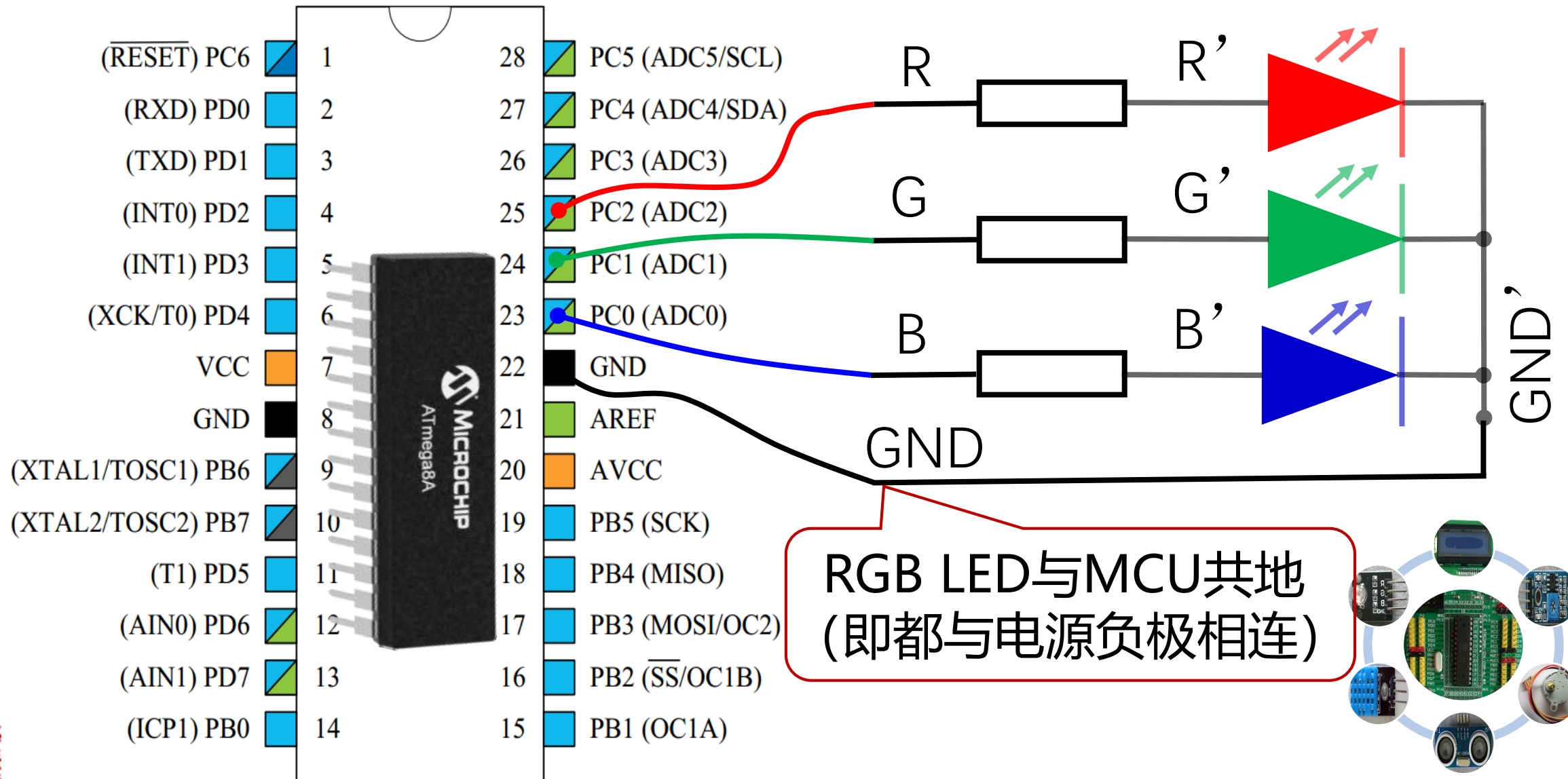
$a \&(1 \ll 2 | 1 \ll 1)$



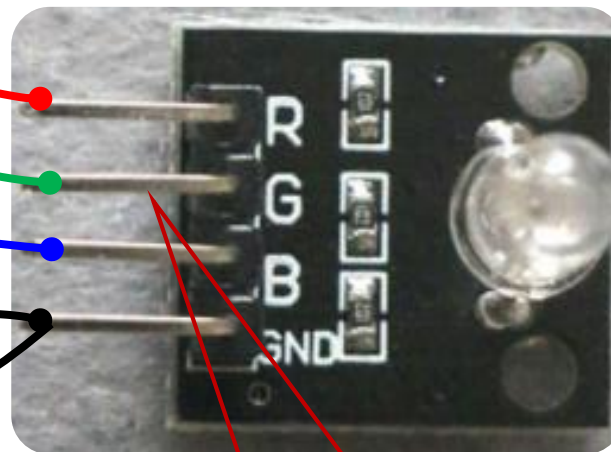
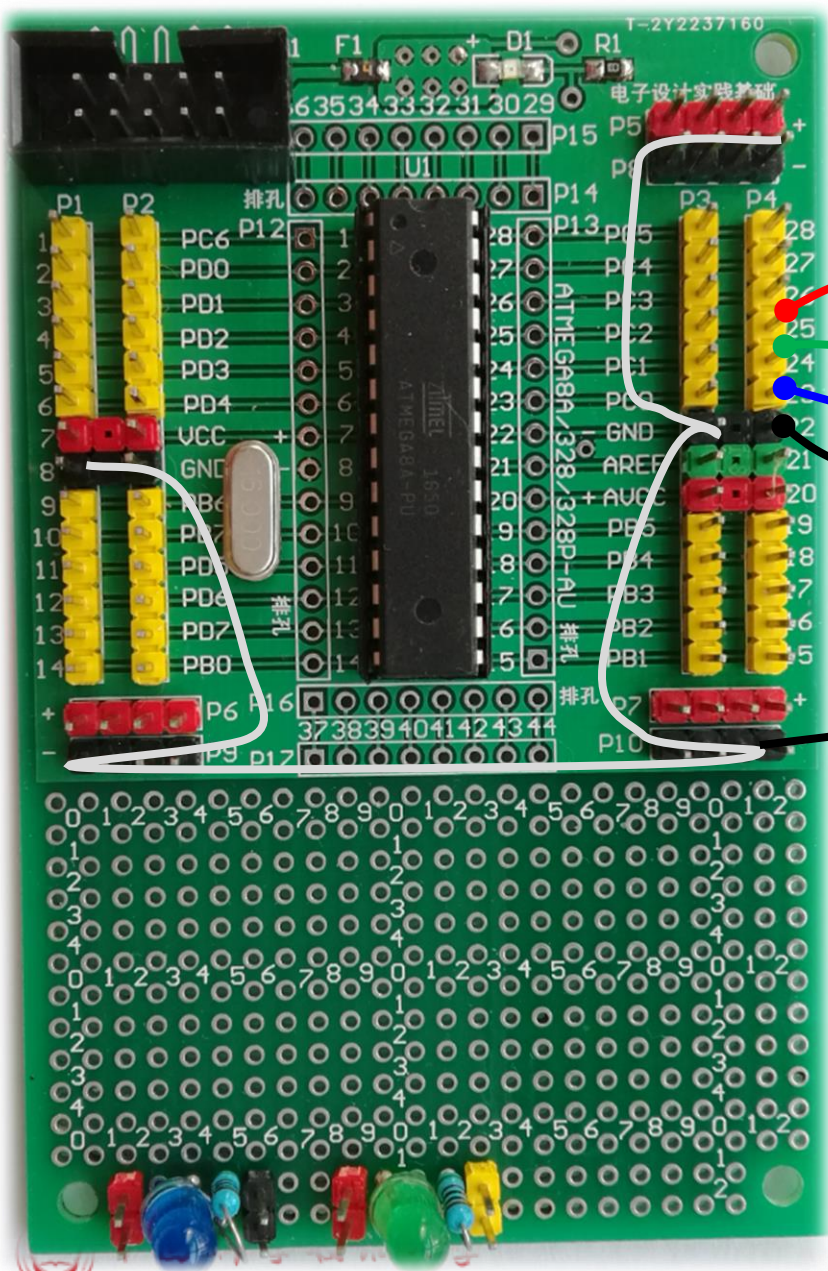
RGB LED模块



连接RGB LED到ATmega8A：原理

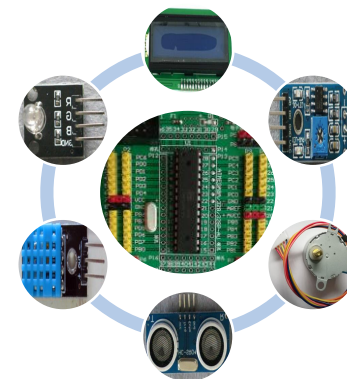


连接RGB LED到ATmega8A: 实物



RGB LED与MCU
共地（即与供电
电源负极相连）

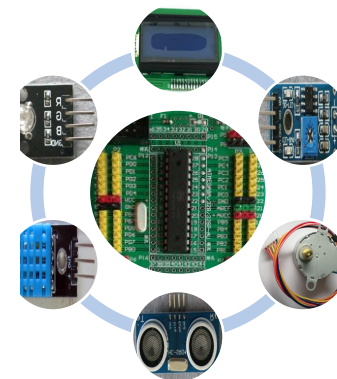
由PC0/1/2
管脚分别控制LED的正
极B/G/R



控制RGB LED编程的关键

```
int i,j;  
DDRC |= (1<<DDRC2) | (1<<DDRC1) | (1<<DDRC0);  
while (1)  
{ PORTC = (1<<PORTC2); //only red on  
  for(i=0;i<100;i++)  
  for(j=0;j<1000;j++);  
  PORTC = (1<<PORTC1); //only green on  
  for(i=0;i<100;i++)  
  for(j=0;j<1000;j++);  
  PORTC = (1<<PORTC0); //only blue on  
  for(i=0;i<100;i++)  
  for(j=0;j<1000;j++);  
}
```

- 流水灯：轮流点亮每一个颜色



编程拓展：延时

```
int i,j;
```

```
DDRC = (1 << DDRC2)|(1 << DDRC1)|(1 << DDRC0);
```

```
while (1)
```

```
{
```

```
PORTC = (1 << PORTC2); //only red on
```

```
for(i=0;i<100;i++)
```

```
for(j=0;j<1000;j++);
```

```
PORTC = (1 << PORTC1); //only green on
```

```
for(i=0;i<100;i++)
```

```
for(j=0;j<1000;j++);
```

```
PORTC = (1 << PORTC0); //only blue on
```

```
for(i=0;i<100;i++)
```

```
for(j=0;j<1000;j++);
```

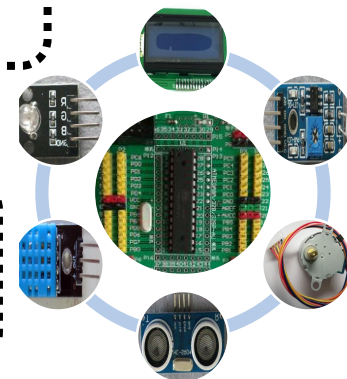
```
}
```

1 `#include <util/delay.h>`

2 `#define F_CPU 1000000UL`

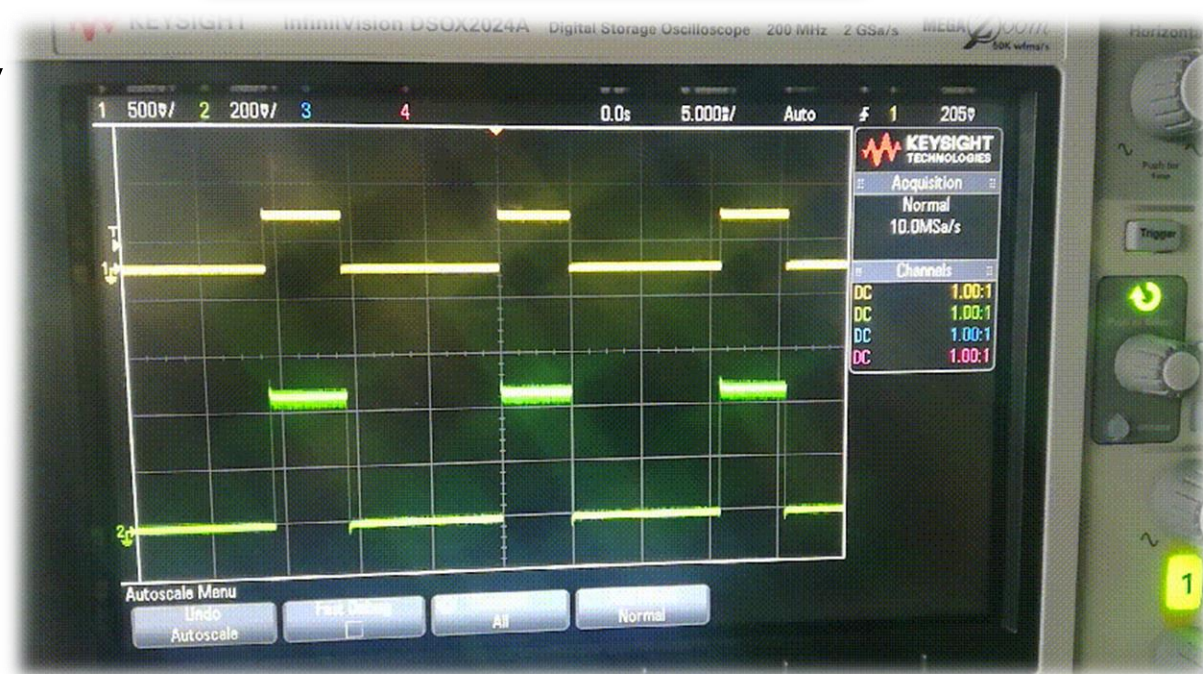
3 `_delay_ms(?);`

`_delay_us(常数);`



知识拓展：LED呼吸灯

- 灯的亮灭，模仿呼吸过程
- 整个过程约3秒，渐亮/渐灭各1.5秒
- 线性或非线性
- MCU实现原理：调整高/低电平脉冲的宽度，PWM（脉冲宽度调制）
- LED余晖效应

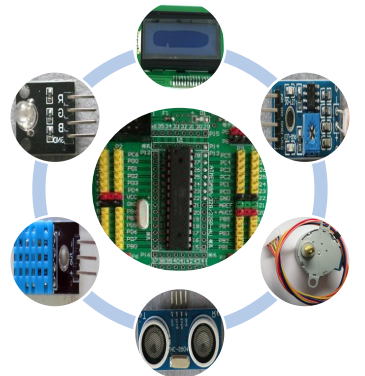


实现呼吸灯的编程关键

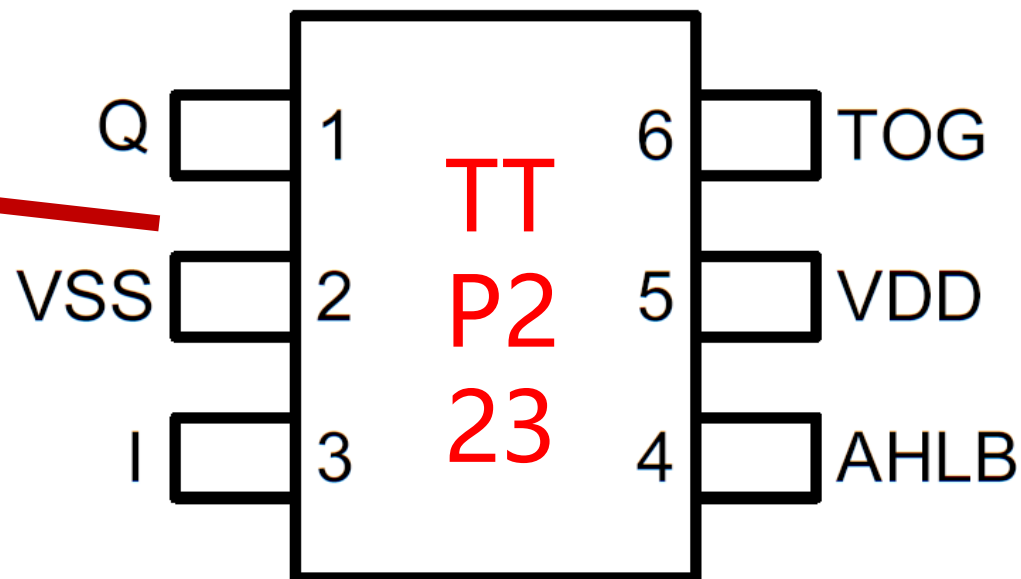
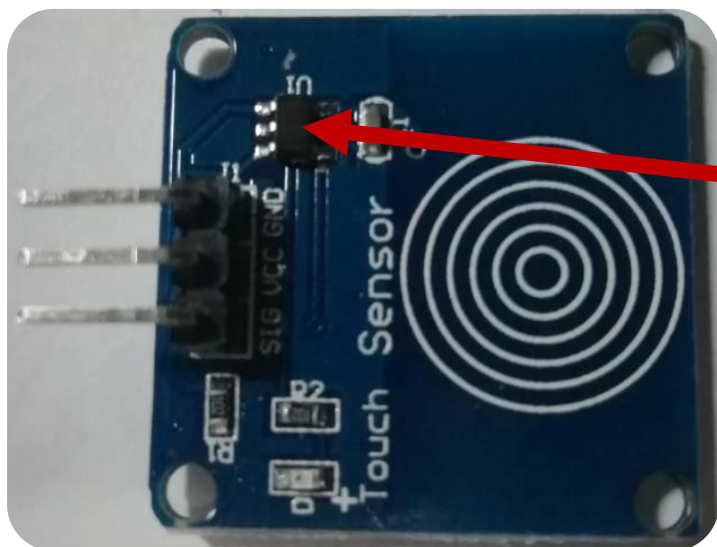
```
int i,j;
DDRC = (1<<DDRC2)|(1<<DDRC1)|(1<<DDRC0); //PC[2~0]为输出
for(i=0;i<100;i++)//渐亮
{ PORTC |= (1<<PORTC0);//PC[0]=1
  for(j=0;j<=i;j++) delay us(150);
  PORTC &= ~(1<<PORTC0);//PC[0]=0
  for(j=0;j<100-i;j++) _delay_us(150);
}
for(i=100;i>0;i--)//渐暗
{ PORTC |= (1<<PORTC0);//PC[0]=1
  for(j=0;j<=i;j++) delay us(150);
  PORTC &= ~(1<<PORTC0);//PC[0]=0
  for(j=0;j<100-i;j++) _delay_us(150);
}
```

- 在PC0管脚外接一个LED，以实现呼吸灯

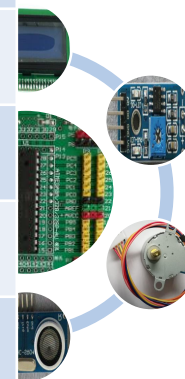
$$150\mu s \times 100 \times 100 \approx 1.5s$$



TTP223 一位触摸开关

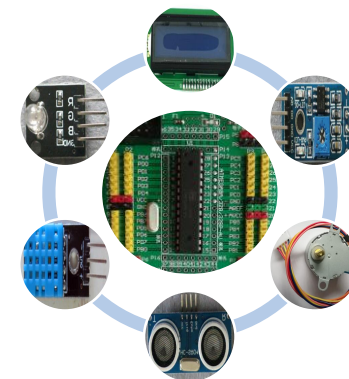
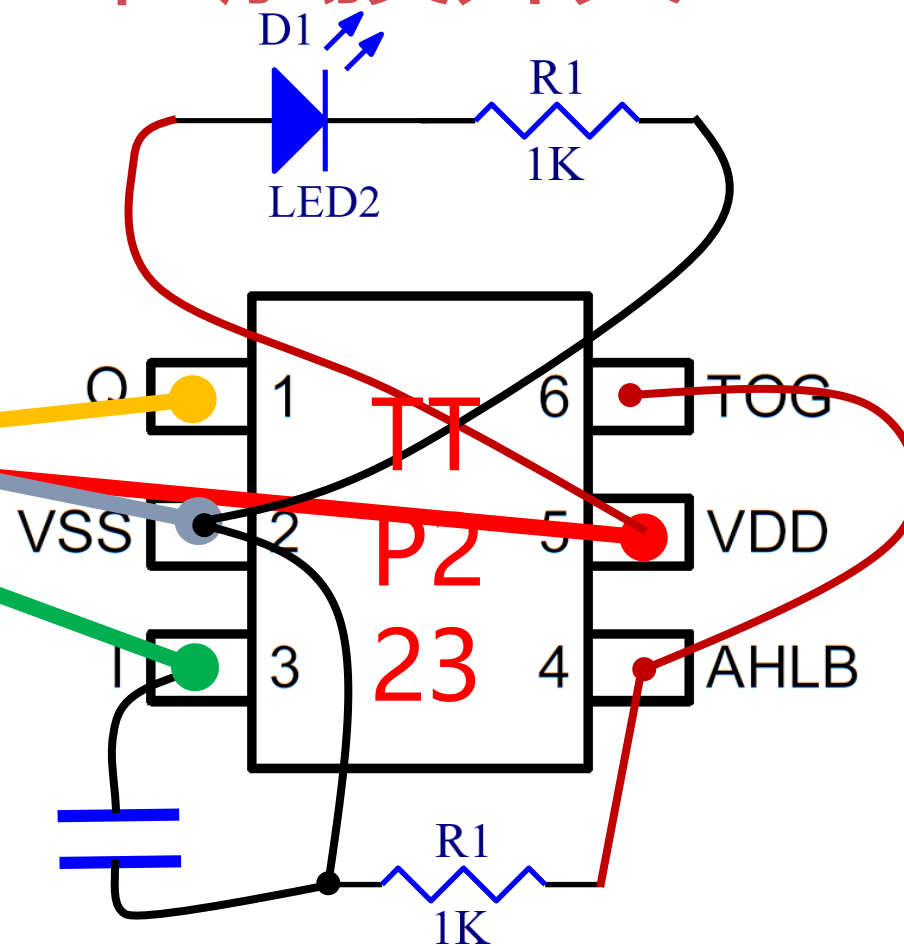
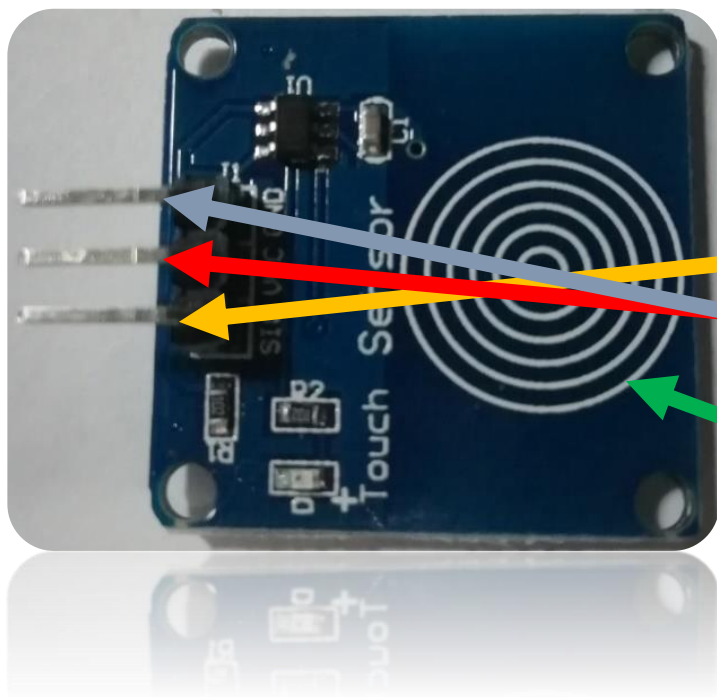


管脚编号	管脚名	IO类型	说明
1	Q	输出	CMOS输出管脚 (1/0)
5/2	VDD/VSS	电源	供电电源正/负极
3	I	输入/出	输入传感器端口
4	AHLB	输入	输出高/低有效电平选择:0-高,1-低
6	TOG	输入	输出类型选择: 0-直接, 1-切换



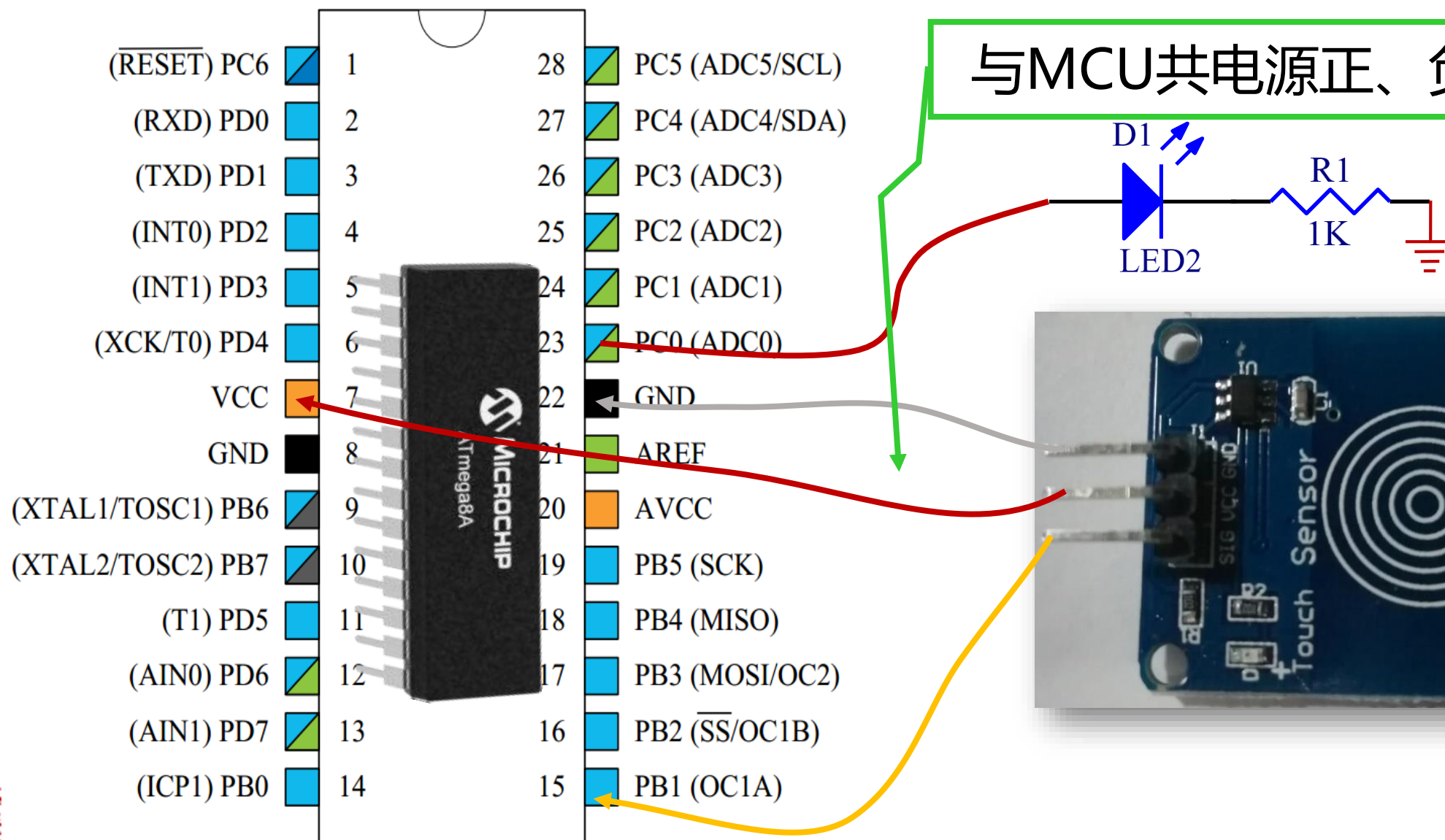
TTP223 一位触摸开关

GND
VCC
SIG

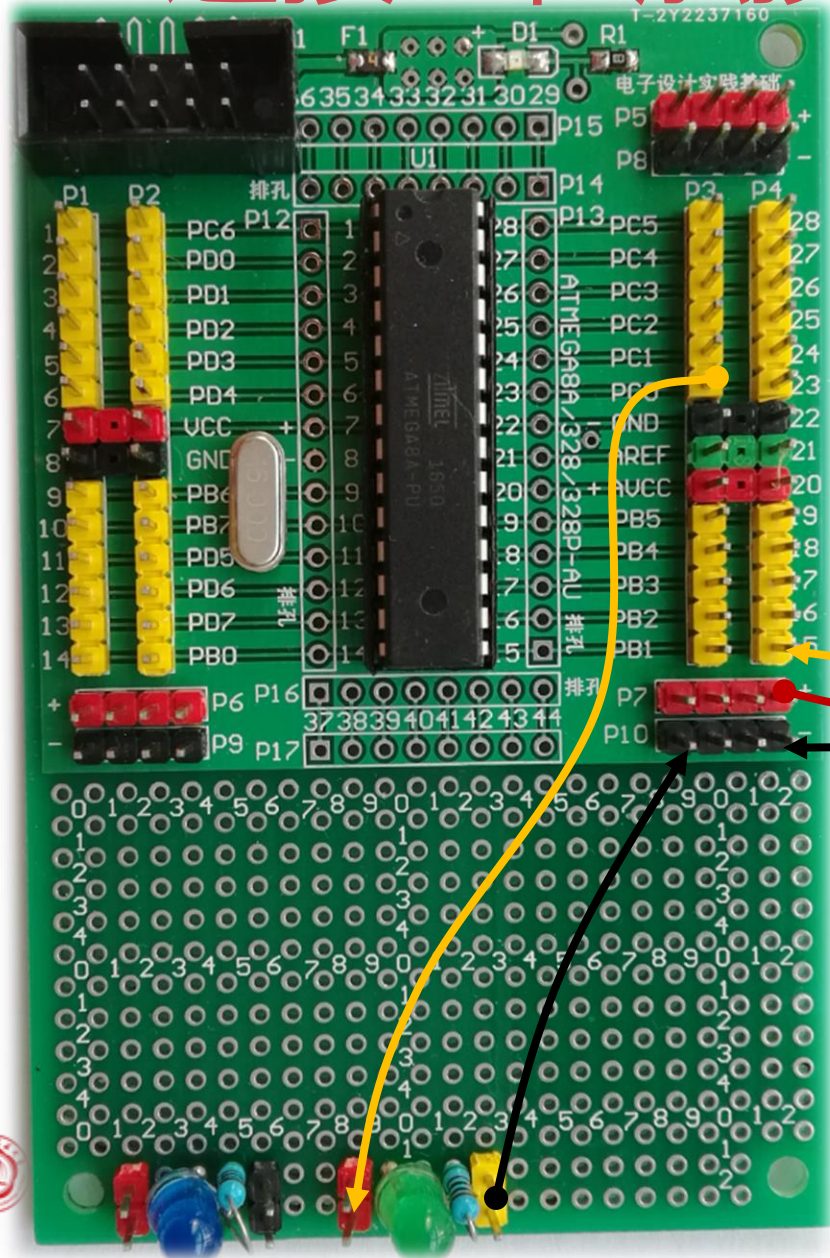


连接一位触摸开关到ATmega8A：原理

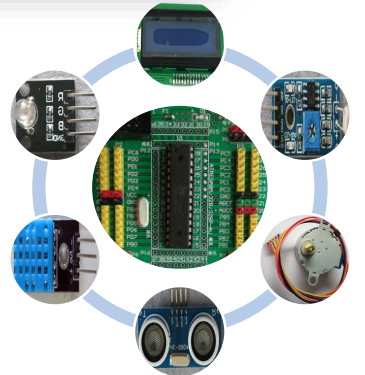
与MCU共电源正、负极



连接一位触摸开关到ATmega8A：实物

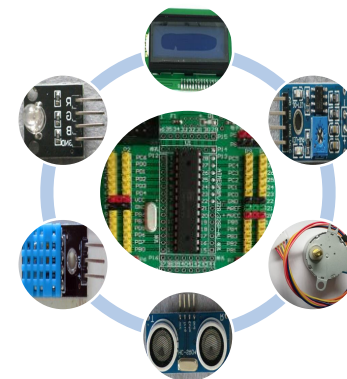


与MCU共电源正、负极

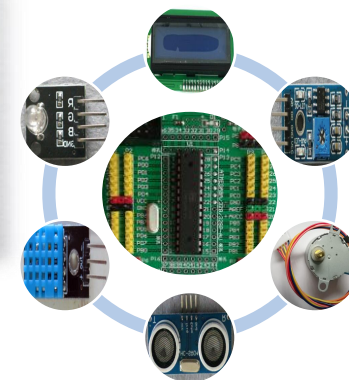
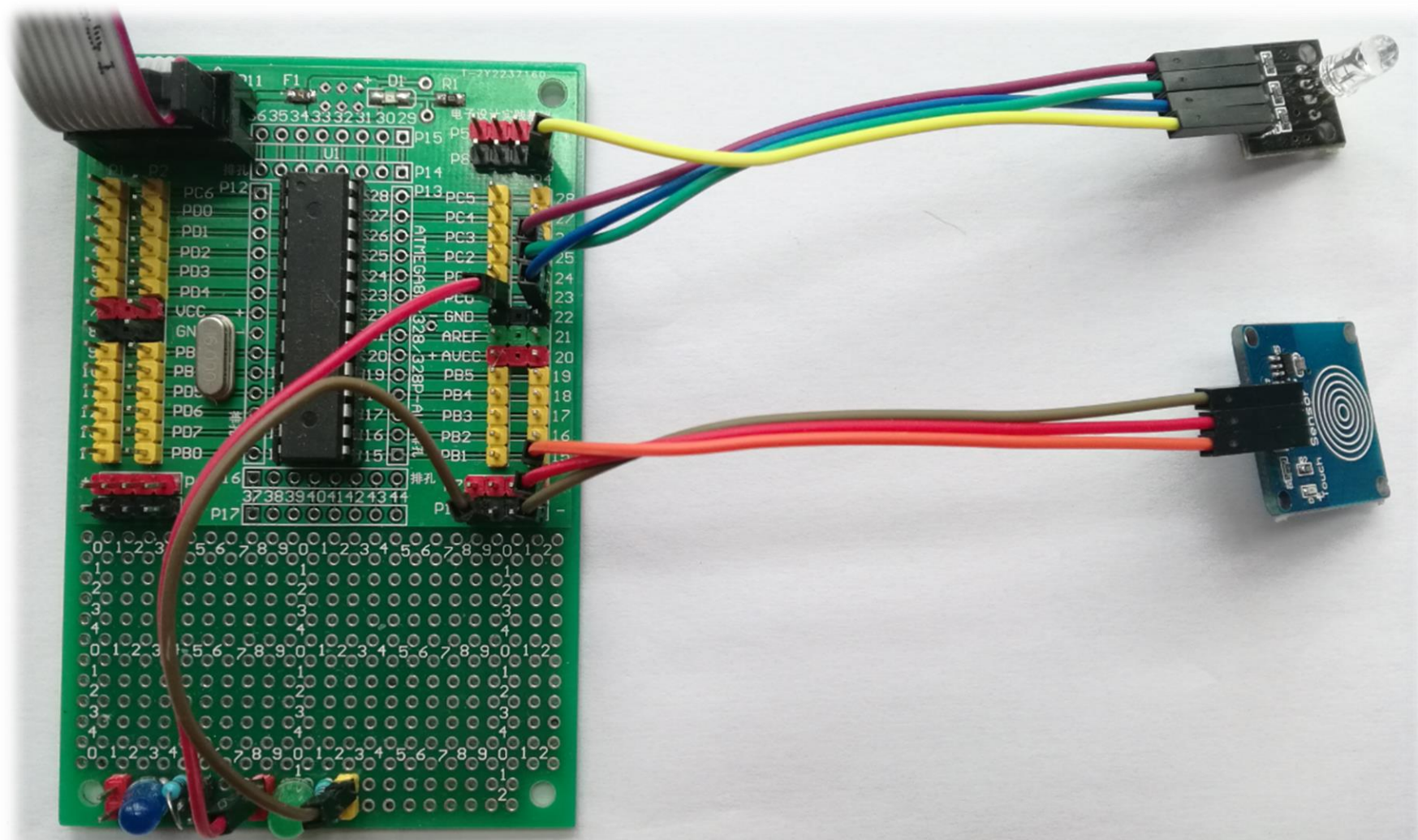


TTP223 一位触摸开关的编程关键

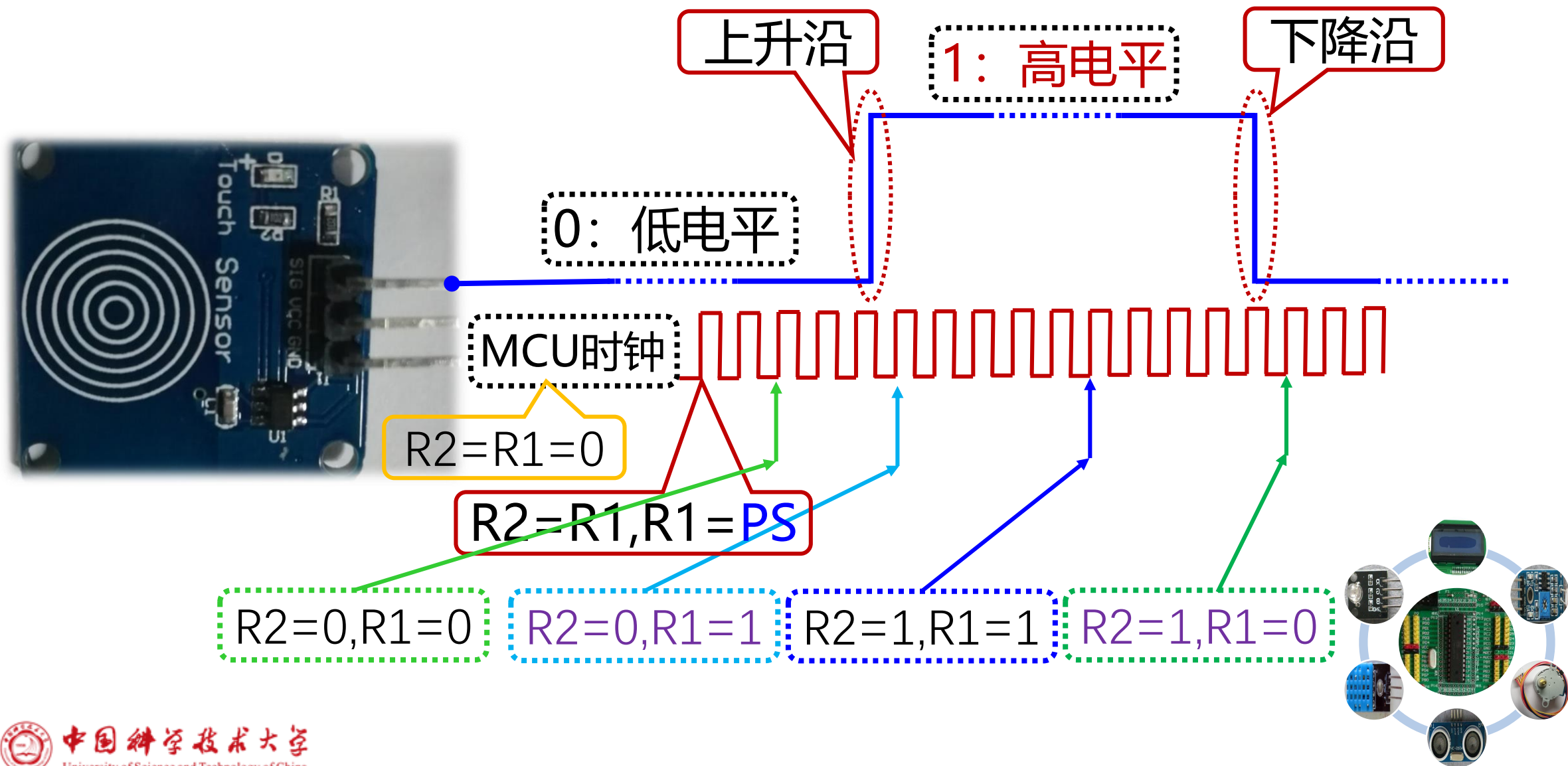
```
DDRC |= (1<<DDRC0); //PC0为输出
DDRB &= ~(1<<DDRB1); //PB1为输入
while (1)
{
    if(PINB & (1<<PINB1)) //PINB[1]=1?
        PORTC |= (1<<PORTC0); //PORTC[0]=1
    else
        PORTC &= ~(1<<PORTC0); //PORTC[0]=0
}
```



一位触摸开关编程的硬件验证



知识拓展：如何统计触摸开关的开关次数



用TTP223控制RGB LED以不同方式亮灭

• 一位触摸开关控制RGB LED的编程关键

```
unsigned char tpr1=0,tpr2=0,counter=0;
```

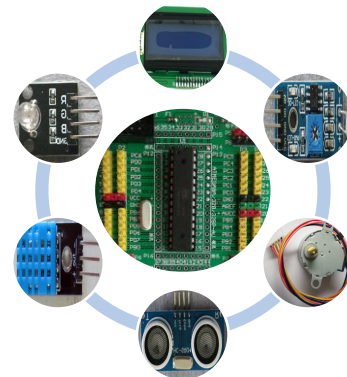
```
DDRC = (1<<DDRC2)|(1<<DDRC1)|(1<<DDRC0); //PC0/1/2为输出模式
```

```
DDRB &= ~(1<<DDRB1); //PB1为输入
```

while(1)

```
    tpr2 = tpr1;
    tpr1 = (PINB & (1<<PINB1))>>1;
    if(tpr2==0 && tpr1==1)counter++;
    switch(counter)
    {
    case 1 : PORTC = (1<<PORTC2);break;
    case 2 : PORTC = (1<<PORTC1);break;
    case 3 : PORTC = (1<<PORTC0);break;
    default :PORTC =0;counter = 0;
    }
```

- 用接在PB1管脚的触摸开关控制PC0/1/2管脚上的RGB LED的亮灭

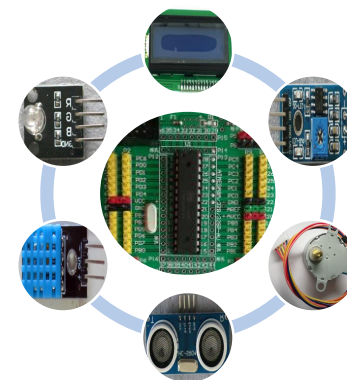


思考

```
tpr2 = tpr1;  
tpr1 = (PINB & (1<<PINB1))>>1;  
if(tpr2==0 && tpr1==1)counter ++;  
switch(counter)  
{  
case 1 : PORTC = (1<<PORTC2);break;  
case 2 : PORTC = (1<<PORTC1);break;  
case 3 : PORTC = (1<<PORTC0);break;  
default :PORTC =0;counter = 0;  
}
```

1, 没有default语句会怎么样?

2, 如何让2个或以上的LED亮



本周实验内容

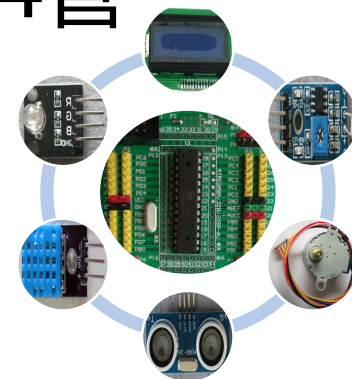
实验内容1：用**PB3/2/1**分别控制RGB LED中的R,G和B正极，实现**三种颜色以上**的循环亮灭

实验内容2：利用**PD2**管脚实现**蓝色的呼吸灯**

实验内容3：用接在**PD1**管脚外的**触摸开关**去控制**PB0**管脚外连接的LED按**呼吸灯**方式亮灭

实验内容4：用接在PD6管脚的触摸开关控制PB2,3,4管脚上的RGB LED的亮灭

当次全部完成后当场验收，总结下次交



实验注意事项与应急处理



- 连接下载线到电脑USB接口时应注意：
 - 实验板电源指示灯是否亮，如不亮可能存在连接短路，须立即断开下载线与电脑USB接口的连接
- 打开下载程序PROGISP (Ver 2.0)
 - 确认下载线连接正确（连接状态图标为彩色）
 - 确认芯片型号正确，调入Flash程序正确等

