



苏州大学

SOOCHOW UNIVERSITY



# 第6章 串行通信模块及第一个中断程序结构

**6.1** 异步串行通信的通用基础知识

**6.2** MSP432芯片UART驱动构件及其使用方法



苏州大学

SOOCHOW UNIVERSITY

arm



## 6.1

# 异步串行通信的 通用基础知识





## 6.1

## 异步串行通信的通用基础知识

### 定义

串行通信接口，即异步串行通信，简称UART，也就是我们常说的“串口”或SCI，在USB未普及之前，是PC机必备的通信接口之一

### 特殊地位

串口通信具有特殊重要地位的原因

- 原因一：在通信方式上，属于单字节通信，是嵌入式开发中重要的调试手段之一
- 原因二：它是最简单的串行通信方式，硬件上所说的232、485通信均是指这个串口通信
- 原因三：硬件接线简单，与MCU引脚直接相连的一般只需要发送线、接收线和地线



## 6.1

# 异步串行通信的基础知识-串行通信的基本概念



1

异步串行通信的格式

2

串行通信的波特率

3

奇偶校验

4

串行通信传输方式术语



## 6.1

### 异步串行通信的基础知识-异步串行通信的格式

串行通信数据以字节为单位，按位的顺序从发送线送出。从MCU引脚的来看，高电平就是逻辑“1”，低电平就是逻辑“0”



#### 串行通信格式

- 串行通信的发送线，平时处于逻辑“1”状态





## 6.1

### 异步串行通信的基础知识-异步串行通信的格式

串行通信数据以字节为单位，按位的顺序从发送线送出。从MCU引脚的来看，高电平就是逻辑“1”，低电平就是逻辑“0”



#### 串行通信格式

- 串行通信的发送线，平时处于逻辑“1”状态
- 开始发送一个字节前，先发送一位逻辑“0”，即起始位，表示发送一个字节的开始



## 6.1

### 异步串行通信的基础知识-异步串行通信的格式

串行通信数据以字节为单位，按位的顺序从发送线送出。从MCU引脚的来看，高电平就是逻辑“1”，低电平就是逻辑“0”



#### 串行通信格式

- 串行通信的发送线，平时处于逻辑“1”状态
- 开始发送一个字节前，先发送一位逻辑“0”，即起始位，表示发送一个字节的开始
- 随后发送一个字节，然后回到平时状态的逻辑“1”



## 6.1

### 异步串行通信的基础知识-异步串行通信的格式

串行通信数据以字节为单位，按位的顺序从发送线送出。从MCU引脚的来看，高电平就是逻辑“1”，低电平就是逻辑“0”



#### 串行通信格式

- 串行通信的发送线，平时处于逻辑“1”状态
- 开始发送一个字节前，先发送一位逻辑“0”，即起始位，表示发送一个字节的开始
- 随后发送一个字节，然后回到平时状态的逻辑“1”
- 每发送一个字节前，均要先发送一个逻辑“0”这个起始位，这就是“异步”之含义，所以称为异步通信





苏州大学

SOOCHOW UNIVERSITY

arm



## 6.1

# 异步串行通信的基础知识-串行通信的基本概念

1

异步串行通信的格式



2

串行通信的波特率

3

奇偶校验

4

串行通信传输方式术语



## 6.1

# 异步串行通信的基础知识-串行通信的波特率

### 波特率 定义

人们把每秒内传送的位数叫做波特率

单位是：位/秒，记为bps

常用的波特率有9600、19200等

### 波特率 基本概念

在包含开始位与停止位的情况下，发送一个字节需10位，很容易计算出，在特定波特率下，发送1K字节所需的时间。波特率越高，位长越小，易受电磁干扰，所以串行通信速度不能很高。距离短时可以适当提高波特率，但提高幅度有限



## 6.1

## 异步串行通信的基础知识-串行通信的波特率

### 课堂习题

- (1) 画出发送数据0x65的串行时序格式图
- (2) 设串行发送引脚高电平为3.3V，低电平为0V，问：  
连续发送字符“A”，发送引脚上的平局电压是多少？



## 6.1

# 异步串行通信的基础知识-串行通信的基本概念

1

异步串行通信的格式

2

串行通信的波特率

3

奇偶校验

4

串行通信传输方式术语







## 6.1

### 异步串行通信的基础知识-奇偶校验

#### 目的

在异步串行通信中，如何知道一个字节的传输是否正确？最常见的方法是**增加一个位**（奇偶校验位），供错误检测使用

#### 定义

字符奇偶校验检查（Character Parity Checking，CPC）称为垂直冗余检查（Vertical Redundancy Checking，VRC），它是为每个字符增加一个**额外位**使字符中“1”的个数为奇数或偶数

奇数或偶数依据使用的是“奇校验检查”还是“偶校验检查”而定。当使用“**奇校验检查**”时，如果字符数据位中“1”的数目是偶数，校验位应为“1”，如果“1”的数目是奇数，校验位应为“0”。当使用“**偶校验检查**”时，如果字符数据位中“1”的数目是偶数，则校验位应为“0”，如果是奇数则为“1”



## 6.1

## 异步串行通信的基础知识-奇偶校验

### 实例

**ASCII字符“R”，其位构成是1010010**

**由于字符“R”中有3个位为“1”，若使用奇校验检查，则校验位为0；如果使用偶校验检查，则校验位为1**

注：几乎所有MCU的串行异步通信接口，都提供这种功能。但实际编程使用较少，原因是单字节校验意义不大



## 6.1

# 异步串行通信的基础知识-串行通信的基本概念

1

异步串行通信的格式

2

串行通信的波特率

3

奇偶校验

4

串行通信传输方式术语





## 6.1 异步串行通信的基础知识-串行通信传输方式术语

### 全双工 ( Full-duplex )

1

数据传送是双向的，且可以同时接收与发送数据。这种传输方式中，除了地线之外，需要两根数据线，站在任何一端的角度看，一根为发送线，另一根为接收线。一般情况下，MCU的异步串行通信接口均是全双工的

### 半双工 ( Half-duplex )

2

数据传送也是双向的，但是在这种传输方式中，除地线之外，一般只有一根数据线。任何时刻，只能由一方发送数据，另一方接收数据，不能同时收发

### 单工 ( Simplex )

3

数据传送是单向的，一端为发送端，另一端为接收端。这种传输方式中，除了地线之外，只要一根数据线就可以了。有线广播就是单工的





## 6.1

## 异步串行通信的通用基础知识-RS232总线标准



为使信号传输得更远，美国电子工业协会制订了串行物理接口标准**RS232**

**异步串行通信格式**：实际的异步串行通信采用的是**NRZ**数据格式，即“标准不归零传号/空号数据格式”。“不归零”是：用负电平表示一种二进制值，正电平表示另一种二进制值，不使用零电平。“传号/空号”分别是表示两种状态的物理名称，逻辑名称记为“1/0”



## 6.1

# 异步串行通信的通用基础知识-RS232总线标准

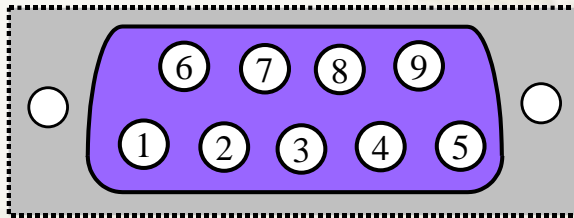
## RS232总线标准

- RS232采用负逻辑
- -15V ~ -3V为逻辑“1”
- +3V ~ +15V为逻辑“0”
- 最大的传输距离是30m
- 通信速率一般低于20Kbps



## 6.1

## 异步串行通信的通用基础知识-RS232总线标准



## 9芯串行接口排列

注：RS232一般使用9芯串行接口

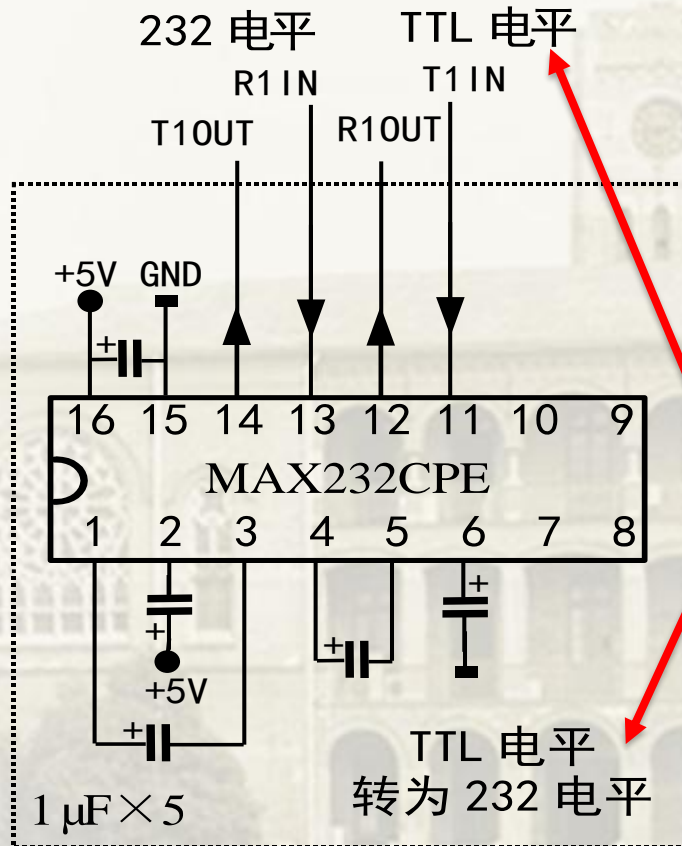
引脚号	功能	引脚号	功能
1	接收线信号检测	6	数据通信设备准备就绪(DSR)
2	接收数据线(RxD)	7	请求发送(RTS)
3	发送数据线(TxD)	8	允许发送(CTS)
4	数据终端准备就绪(DTR)	9	振铃指示
5	信号地(SG)		

注：在RS232通信中，常常仅使用3根线：接收线、发送线和地线



## 6.1

# 异步串行通信的通用基础知识-TTL电平到RS232电平转换电路



在发送端，需要用驱动电路将  
TTL电平转换成RS232电平

在接收端，需要用接收电路将  
RS232电平转换为TTL电平

串口通信接口电平转换电路





## 6.1

### 异步串行通信的通用基础知识-TTL电平到RS232电平转换电路

随着USB接口的普及，9芯串口正在逐渐从PC机，特别是从便携式电脑上消失。于是出现232-USB转换线、TTL-USB转换线，在PC机上安装相应的驱动软件，就可在PC机上使用一般的串行通信编程方式，通过USB接口实现与MCU的串行通信



## 6.1 异步串行通信的通用基础知识-串行通信编程模型

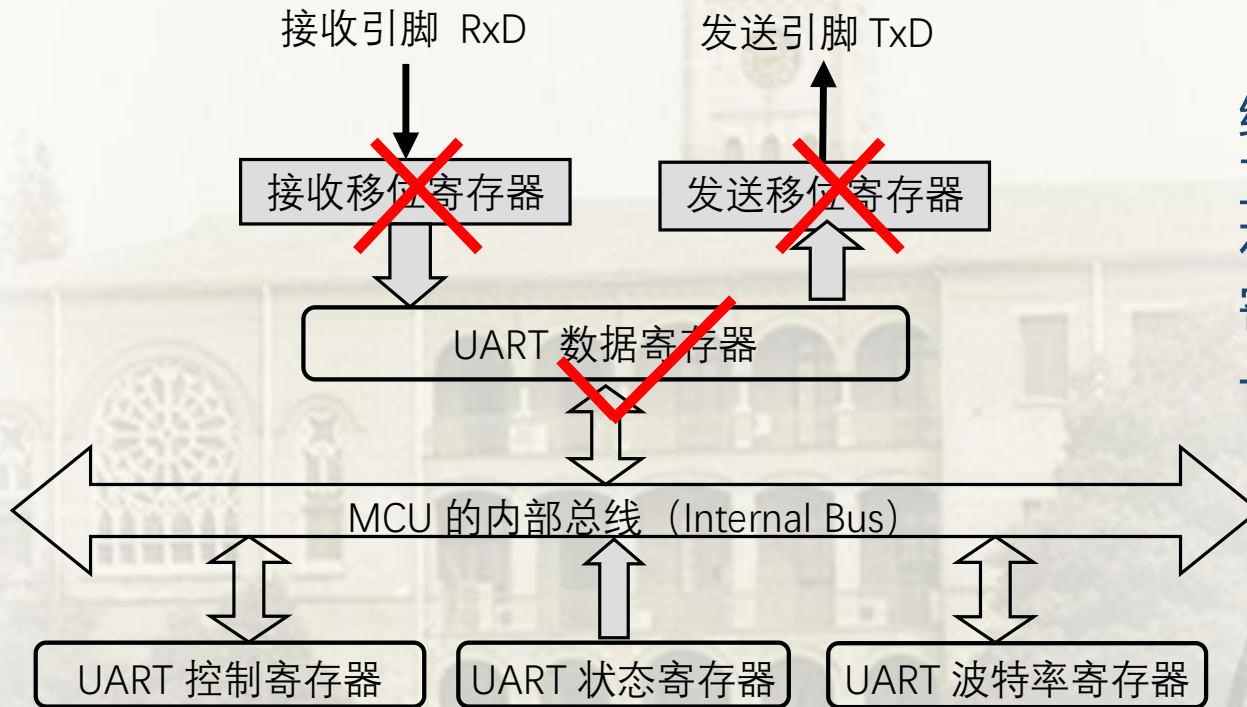
串行通信接口主要功能

接收时，把外部的单线输入的数据变成一个字节的并行数据送入MCU内部

发送时，把需要发送的一个字节的并行数据转换为单线输出



## 6.1 异步串行通信的通用基础知识-串行通信编程模型

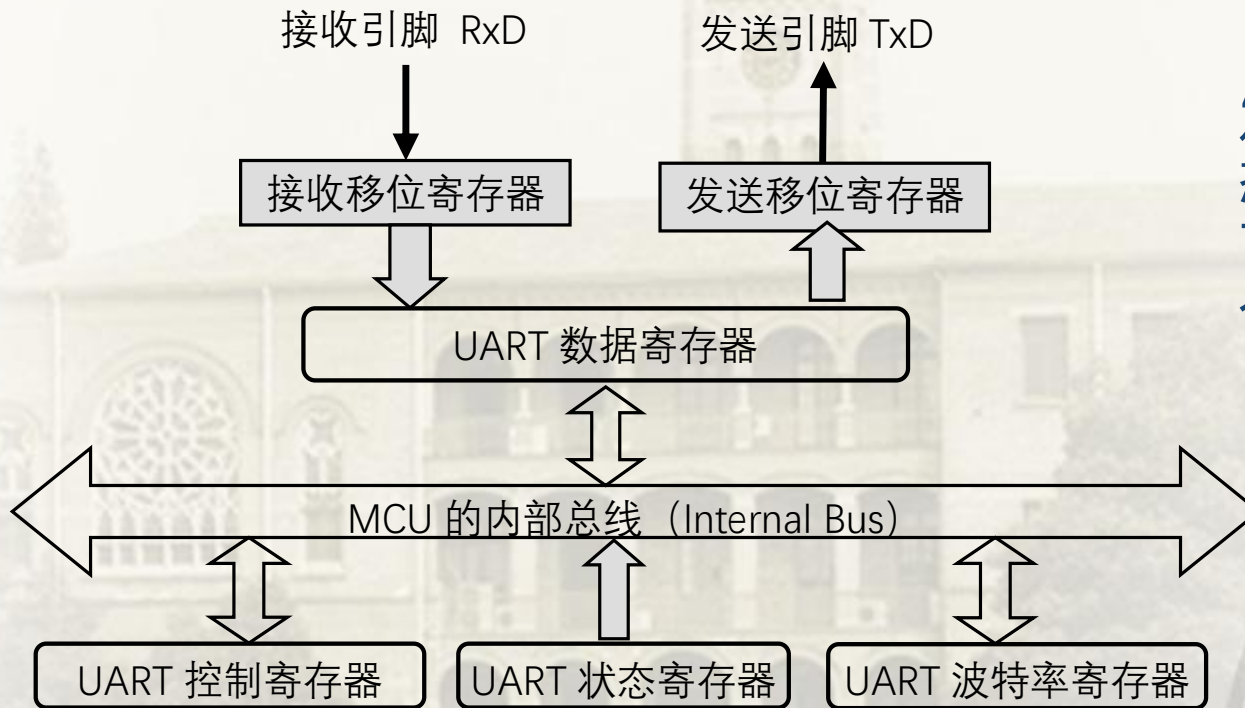


编程时，程序员并不直接与“发送移位寄存器”和“接收移位寄存器”打交道，只与数据寄存器打交道

UART编程模型



## 6.1 异步串行通信的通用基础知识-串行通信编程模型



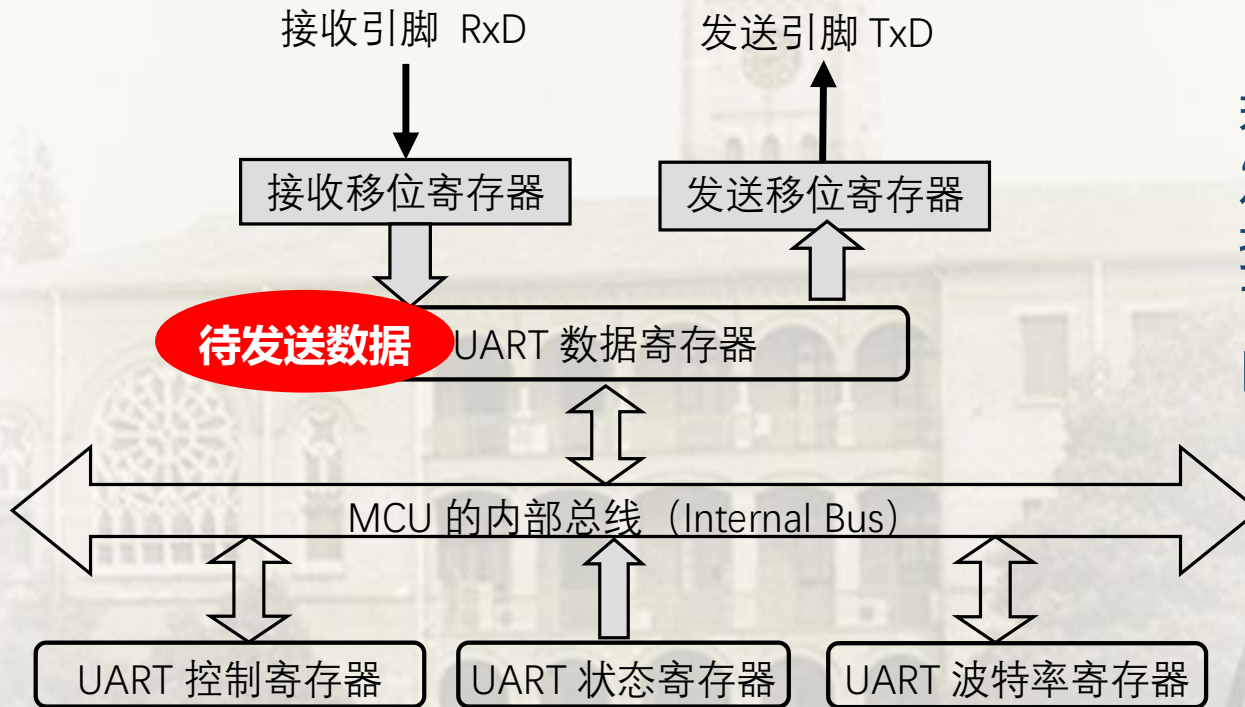
发送时，通过判定状态寄存器的相应位，了解是否可以发送一个新的数据

UART编程模型





## 6.1 异步串行通信的通用基础知识-串行通信编程模型

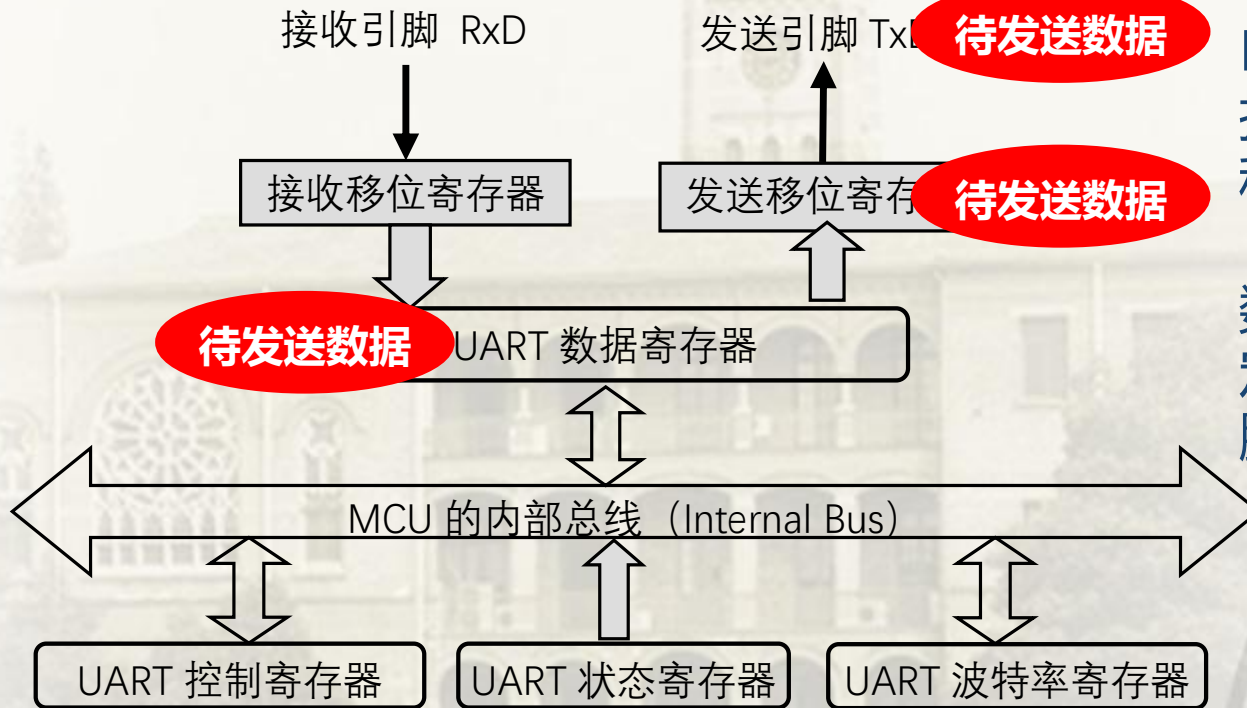


若可以发送，则将待发送的数据放入“数据寄存器”中就可以了，剩下的工作由MCU自动完成

UART编程模型



## 6.1 异步串行通信的通用基础知识-串行通信编程模型

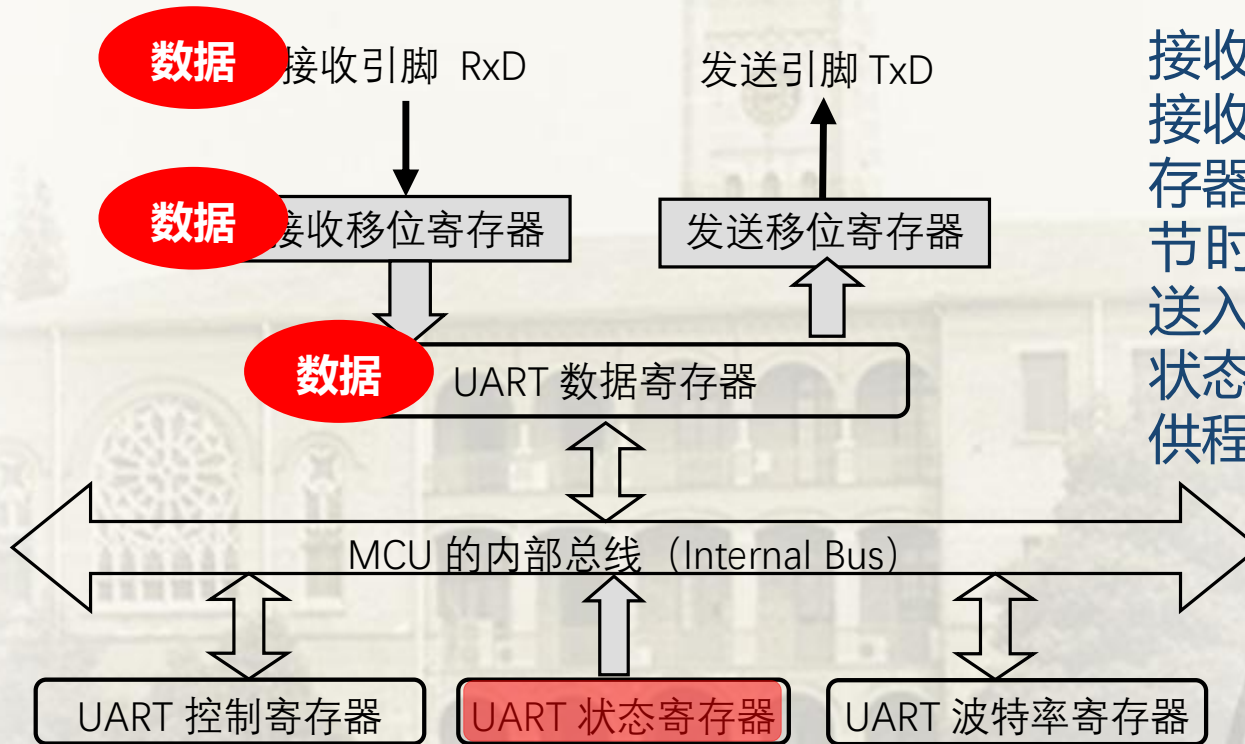


MCU 会将数据从“数据寄存器”送到“发送移位寄存器”，硬件将“发送移位寄存器”的数据一位一位地按照规定的波特率移到发送引脚，供对方接收

UART编程模型



## 6.1 异步串行通信的通用基础知识-串行通信编程模型



接收时，数据一位一位地从接收引脚进入“接收移位寄存器”，当收到一个完整字节时，MCU会自动将数据送入“数据寄存器”，并将状态寄存器的相应位改变，供程序员判定并取出数据

UART编程模型



苏州大学

SOOCHOW UNIVERSITY

arm



6.2

# MSP432芯片UART 驱动构件及使用方法







## 6.2 MSP432芯片UART驱动构件及使用方法-MSP432芯片UART引脚

引脚号	引脚名	引脚功能2
6	P1.2	UCA0RXD
7	P1.3	UCA0TXD
18	P2.2	PM_UCA1RXD
19	P2.3	PM_UCA1TXD
34	P3.2	PM_UCA2RXD
35	P3.3	PM_UCA2TXD
98	P9.6	UCA3RXD
99	P9.7	UCA3TXD

MSP432中共有4个UART模块，分别为eUSCI\_A0、eUSCI\_A1、eUSCI\_A2和eUSCI\_A3。每个UART(以下以UART简称UART模式下的eUSCI\_A模块)的发送数据引脚为UCAxTXD，接收数据引脚为UCAxRXD。“x”表示串口模块编号，取值为0~3



## 6.2

# MSP432芯片UART驱动构件及使用方法-基本要素分析与头文件

串口驱动构件的组成

01

头文件uart.h

02

源代码文件uart.c

供应用程序开发调用





6.2

MSP432芯片UART驱动构件及使用方法-基本要素分析与头文件

串口的三种基本操作

01

初始化

02

发送

03

接收



## 6.2

## MSP432芯片UART驱动构件及使用方法-基本要素分析与头文件

### 串口初始化函数的参数：

**首先**必须确定使用哪个串口；**其次**必须确定使用什么速度收发，这样，串口初始化函数就两个参数：**串口号与波特率**。但是，KL系列MCU的一个串口，可以在不同引脚组上，实际应用中确定使用哪个引脚组，应该是在应用开发板硬件设计阶段就确定的，为了使驱动构件适应这个场景，可在头文件中使用“**宏**”进行定义，确定使用的引脚组





## 6.2

## MSP432芯片UART驱动构件及使用方法-基本要素分析与头文件

串口初始化函数的参数：

从知识要素角度，进一步分析串口驱动构件的基本函数，还应当包含发送单个多个字节、接收单个多个字节的函数，以及使能及禁止接收中断、获取接收中断状态的函数

除了使用串口驱动构件中封装的API函数之外，还可以使用格式化输出函数printf来灵活地从串口输出调试信息，配合PC机或笔记本电脑上的串口调试工具，可方便地进行嵌入式程序的调试，具体使用方法参见下一页



## 6.2

## MSP432芯片UART驱动构件及使用方法-printf的设置方法与使用

### 设置步骤

1、在printf头文件..\ 05\_SoftComponent\printf\printf.h中宏定义需要与printf相关联的调试串口号，例如：

```
#define UART_Debug UART_0 //printf函数使用的串口号
```

2、在使用printf前，调用UART驱动构件中的初始化函数对使用的调试串口进行初始化，配置其波特率。例如：

```
uart_init (UART_Debug,9600); //初始化"调试串口"
```

注：关于printf函数的使用方法，请参见附录D的介绍



苏州大学

SOOCHOW UNIVERSITY

arm

谢谢!

