# 串口通信

串口通信特点：异步、电平信号、串行

波特率：串口通信时每秒钟可以传输的通信位

串口通信的波特率是不能随便定的，而是从一些值中去选

一般最常见的是9600或者115200

一个通信单位是由：起始位+数据位+奇偶校验位+停止位

串口通信一次发送多少位有效数据是可以设定的（一般可选位有6、7、8、9, 99%都是使用8位，因为一般串口通信都是用的ASCII码，ASCII码一个字符正好编码为8位）

串口通信的基本原理

三根通信线 Rx Tx GND

一般开发板都会引出SoC上串口引脚直接输出的TTL电平的串口（X210开发板没有）插座用插针式插座，每个串口引出的都有3根线（Tx、Rx、GND），可以用这些插座直接连接外部的TTL电平的串口设备。

DB9接口

DB9接口有9根通信线，其中3根很重要，为GND，Tx，Rx，必不可少，剩余6根和流控有关，现在我们使用串口都是用来做调试一般都禁用流控，所以这6根没用，软件的设置数据流控制要去掉，不然会出错。

S5PV210串行通信接口详解

串口 uart universial asynchronous receiver transmitter

整个串口控制器包含transmitter和receiver两部分，两部分功能彼此独立，transmitter负责发送信息，receiver负责接收信息

总线角度来讲：串口控制器是接在APB总线上的，利用APB总线上的时钟

transmitter是由发送缓冲区和移位器构成，发送信息时，首先将信息编码（ASCII）成二进制流，然后将一帧数据（一般是8位）写入发送缓冲区，（从这里程序就不用管了，剩下的发送部分是硬件自动的），发送移位器会自动从发送缓冲区读取一帧数据，然后自动移位，（移位的目的是将一帧数据的各个位分别拿出来）将其发送到Tx通信线上

receiver和transmitter相反，接受完一帧数据后receiver会产生一个中断个CPU，来取数据。

总结：程序员只需要操纵发送缓冲区和接收缓冲区（串口的寄存器），其他都是自动的

串口控制中还有个波特率发生器，作用是产生串口发送/接受的节拍时钟。波特率发生器就是个时钟分频器，他的工作需要源时钟（APB总线），然后内部将源时钟进行分频（软件设置寄存器来配置）得到目标时钟，然后再用目标时钟产生波特率（硬件自动）

自动流控：

为什么需要流控？发送方和接收方速度不匹配，流控线通知对方是否处理完毕

现在为什么不用流控？USB、internet的通讯方式，串口只用作SoC输出调试信息的。发送方和接收方硬件都很强，串口已经没有那么快。

高级串口

FIFO模式及作用

典型的串口设计：发送/接收缓冲区只有1字节，每次发送/接受只能处理一帧数据。这样在单片机设计中没什么问题。

DMA模式

IrDA模式

红外线通信（电视机，空调遥控器）

因为串口通信是异步的，异步的意思就是说发送方占主导权，也就是说发送方随时想发就能发，但是接受方只有时刻等待才不会丢失数据，所以这个差异就导致发送方可以不使用中断，而接受方必须使用中断。

串口通信中时钟的设置主要看寄存器设置。重点的有：寄存器源设置（为串口控制其选择源时钟，一般选择为PCLK\_PSYS,也可以是SCLK\_UART）

波特率发生器有两个重要寄存器：UBRDIVn和UNDIVSLOTn，其中UBRDIVn是主要的设置波特率的寄存器，UDIVSLOTn是用来辅助设置的，目的是为了校准波特率的。

main.c

//这里的main.c知识一个函数，和c语言中的main函数不一样。

void main.c(void){

uart\_init();

uart\_putc('a');

uart\_getc();

}

串口的初始化步骤：

1. 初始化串口的Tx和Rx引脚所对应的GPIO（查原理图可知Rx和Tx分别对应GPA0\_1和GPA0\_0）
2. GPA0CON(0xE0200000),bit[3:0]=0b0010 bit[7:4]=0b0010
3. 关键寄存器UCON ULCON UMCON UFCON UBRDIV0 UDIVSLOT0

主要的几个寄存器

ULCON0=0x3 //0校验位、8数据位、1停止位

UMCON0=0x0 //发送和接收都是polling mode 轮询模式

UMCON0=0x0 //禁止modem、afc 流控作用的

UFCON0=0x0 //禁止FIFO模式

UBRDIV0和UDIVSLOT0和波特率有关，要根据公式去计算。

在C源文件中定义访问寄存器的宏

uart.c

#define GPA0CON 0xE0200000

#define UCON0 0xE2900004

#define ULCON0 0xE2900000

#define UMCON0 0xE290000C

#define UFCON0 0xE2900008

#define UBRDIV0 0xE2900028

#define UDIVSLOT0 0xE290002C

#define UTRSTART0 0xE2900010

#define UTXH0 0xE2900020

#define URXH0 0xE2900024

#define rGPA0CON (\*(volatile unsigned int \*) GPA0CON)

#define rUCON0 (\*(volatile unsigned int \*) UCON0)

#define rULCON0 (\*(volatile unsigned int \*) ULCON0)

#define rUMCON0 (\*(volatile unsigned int \*) UMCON0)

#define rUFCON0 (\*(volatile unsigned int \*) UFCON0)

#define rUBRDIV0 (\*(volatile unsigned int \*) UBRDIV0)

#define rUDIVSLOT0 (\*(volatile unsigned int \*) UDIVSLOT0)

#define rUTRSTART0 (\*(volatile unsigned int \*) UTRSTART0)

#define rUTXH0 (\*(volatile unsigned int \*) UTXH0)

#define rURXH0 (\*(volatile unsigned int \*) URXH0)