

目录

第 3 章	串口(UART)	2
3.1	UART 概述	2
3.1.1	补充(常用通信协议分类及其特征介绍)	2
3.1.2	UART 接口	3
3.1.3	UART 数据帧格式	3
3.1.4	UART 四要素	3
3.2	STM32 的 UART	4
3.2.1	STM32 的 UART 介绍	4
3.2.2	STM32 的串口模块工作流程（重点）	4
3.2.3	STM32 的 UART 特征	4
3.3	STM32 的 UART 框架（重点）	5
3.3.1	框架分析	5
3.4	STM32 的 UART 寄存器	6
3.5	STM32 的 UART 实验	6
3.5.1	硬件分析	6
3.5.2	软件设计	7
3.6	课后作业	8

第3章 串口(UART)

3.1 UART 概述

uart: 串口 是一种通信协议

通信协议: 两个设备或者芯片之间进行数据交流的一种规则

单片机与单片机之间、单片机与 PC 之间

通信协议是单片机最基础但最重要的

学完单片机: uart、iic、spi。(80%)

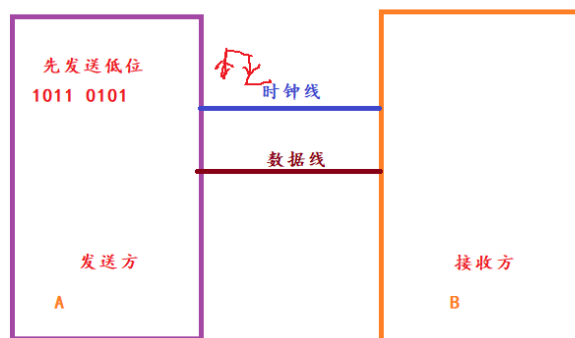
3.1.1 补充(常用通信协议分类及其特征介绍)

1. 同步通信与异步通信

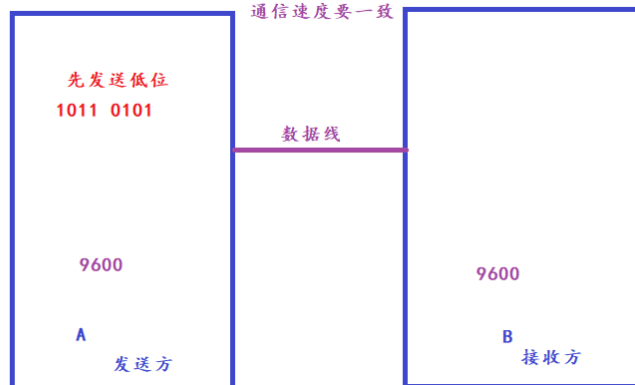
同步: 两个设备或者芯片之间进行数据交流时, 步调要保持一致, 它们之间有一根时钟线相连。

异步: 两个设备或者芯片之间进行数据交流时, 通信速度要一致, 它们之间没有时钟线相连。

时钟线都是在控制着数据线的收发
同步通信 至少都有两根线
时钟线产生边沿变化时, 数据线的收发
假设:
时钟线产生上升沿, 发送方发送数据
时钟线产生下降沿, 接收方采集数据



异步通信: 至少有1根线
通信速度要一致



波特率: 通信速度
9600
表示1S能发9600位数据
发送每一位数据需要1/9600S

2. 单工、半双工、全双工

单工: 两个设备或者芯片之间进行通信时, 发送方和接收方固定, 方向单一。

半双工: 两个设备或者芯片之间进行通信时, 发送方和接收方不固定, 方向是双向, 但是同一时刻只有一个方向。

全双工: 两个设备或者芯片之间进行通信时, 发送方和接收方不固定, 方向是双向, 但是同一时刻可以有两个方向。

3. 串行和并行

串行: 八辆车, 过单行道, 只能一辆接一辆

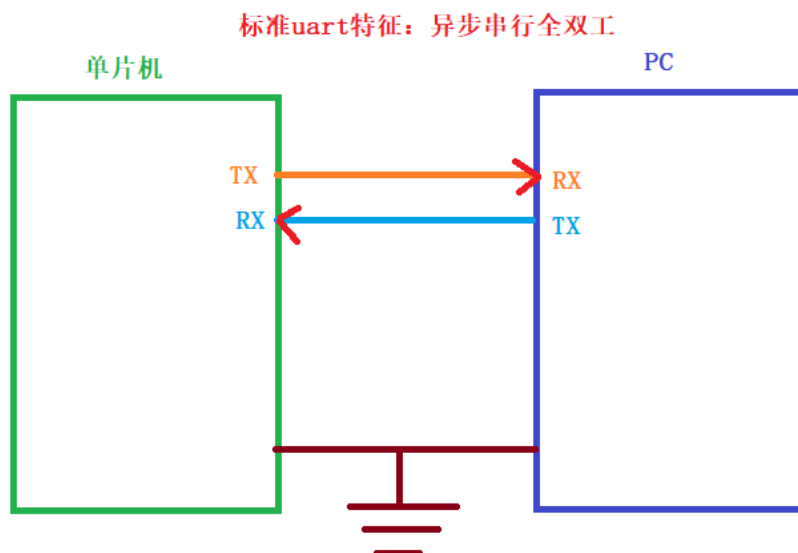
并行: 八辆车, 过八车道, 一次可以通过

4. 板级总线和现场总线

板级总线: 在 PCB 板上面的线

现场总线: 现场拉出来 CAN、485

3.1.2 UART 接口



3.1.3 UART 数据帧格式(重点)



空闲电平：高电平(没有进行通信时)

起始位：一个位的低电平，目的告诉接收方准备接收数据。

数据位：5 – 8bit 位 先低后高

校验位：奇偶校验 可有可无

奇校验：统计前面数据位“1”的个数，0110 1101，校验位为 0，0110 1100，校验位为 1

偶校验：统计前面数据位“1”的个数，0110 1101，校验位为 1，0110 1100，校验位为 0

校验的准确率 50% CRC：循环冗余校验

停止位：0.5-2bit 位的高电平时间，告诉接收方数据发送完毕。

3.1.4 UART 四要素(非常重要)

掌握了 uart 四要素，就相当于掌握了 uart。后面配置都是按照 uart 四要素来进行配置

波特率	115200
停止位	1
数据位	8
奇偶校验	无

波特率：通信速度，1S 发送 115200 位数据 发送一位数据的时间：1/115200 S

停止位：0.5-2bit 通信双方保持一致

数据位：5 – 8bit 通信双方保持一致

奇偶校验：可有可无 无

3.2 STM32 的 UART

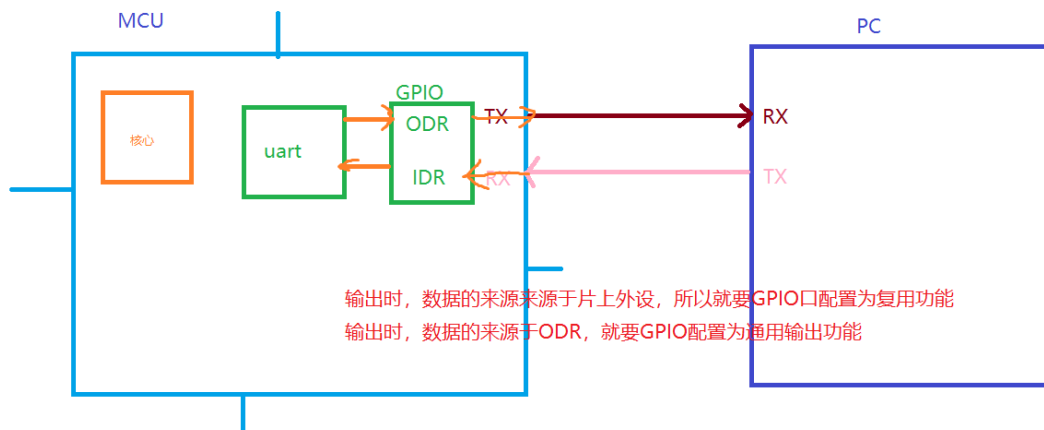
3.2.1 STM32 的 UART 介绍

通用同步异步收发器 (USART) 能够灵活地与外部设备进行全双工数据交换, 满足外部设备对工业标准 NRZ 异步串行数据格式的要求。USART 通过小数波特率发生器提供了多种波特率(多种通信速度)。

它支持同步单向通信和半双工单线通信; 还支持 LIN (局域互连网络)、智能卡协议与 IrDA (红外线数据协会) SIR ENDEC 规范, 以及调制解调器操作 (CTS/RTS)。而且, 它还支持多处理器通信。

通过配置多个缓冲区使用 DMA(直接存储器访问) 可实现高速数据通信。

3.2.2 STM32 的串口模块工作流程 (重点)



3.2.3 STM32 的 UART 特征

- 全双工异步通信
- NRZ 标准格式 (标记/空格) 编码格式
- 可配置为 16 倍过采样或 8 倍过采样(接收时), 因而为速度容差与时钟容差的灵活配置提供了可能
16 倍过采样: 波特率不能设太高 8 倍过采样: 波特率设高点 M 以上才算高
- 小数波特率发生器系统 (通信速度)
— 通用可编程收发波特率 (有关最大 APB 频率时的波特率值, 请参见数据手册)。
- 数据字长度可编程 (8 位或 9 位)
- 停止位可配置 - 支持 0.5 - 2 个停止位
- LIN 主模式同步停止符号发送功能和 LIN 从模式停止符号检测功能
— 对 USART 进行 LIN 硬件配置时可生成 13 位停止符号和检测 10/11 位停止符号
- 用于同步发送的发送器时钟输出
- IrDA SIR 编码解码器
— 正常模式下, 支持 3/16 位持续时间
- 智能卡仿真功能
— 智能卡接口支持符合 ISO 7816-3 标准中定义的异步协议智能卡
— 智能卡工作模式下, 支持 0.5 或 1.5 个停止位
- 单线半双工通信
- 使用 DMA (直接存储器访问) 实现可配置的多缓冲区通信
— 使用 DMA 在预留的 SRAM 缓冲区中收/发字节
- 发送器和接收器具有单独使能位
- 传输检测标志位:
— 接收缓冲区已满 会有一个标志位置起来置一 让我们知道 uart 收到数据
— 发送缓冲区为空 也会有一个标志位置起来 让我们知道 uart 可以发送数据
— 传输结束标志 也会有一个标志位置起来 告诉我们通信结束

- 深圳信盈达科技有限公司 专业提供 M4、A9、LINUX、Android 技术服务、技术培训 第 5 页 共 8 页

$$\text{USARTDIV} = \frac{f_{\text{CK}}}{8 \times (2 - \text{OVER8}) \times \text{Tx/Rx 波特率}}$$

未知量 已知量 已知量 已知量

USARTDIV = 84000000.0/(16 * 9600) = 546.875;

MAN_DIV = USARTDIV;//取整 546

FRA_DIV = (USARTDIV - MAN_DIV) * 16;

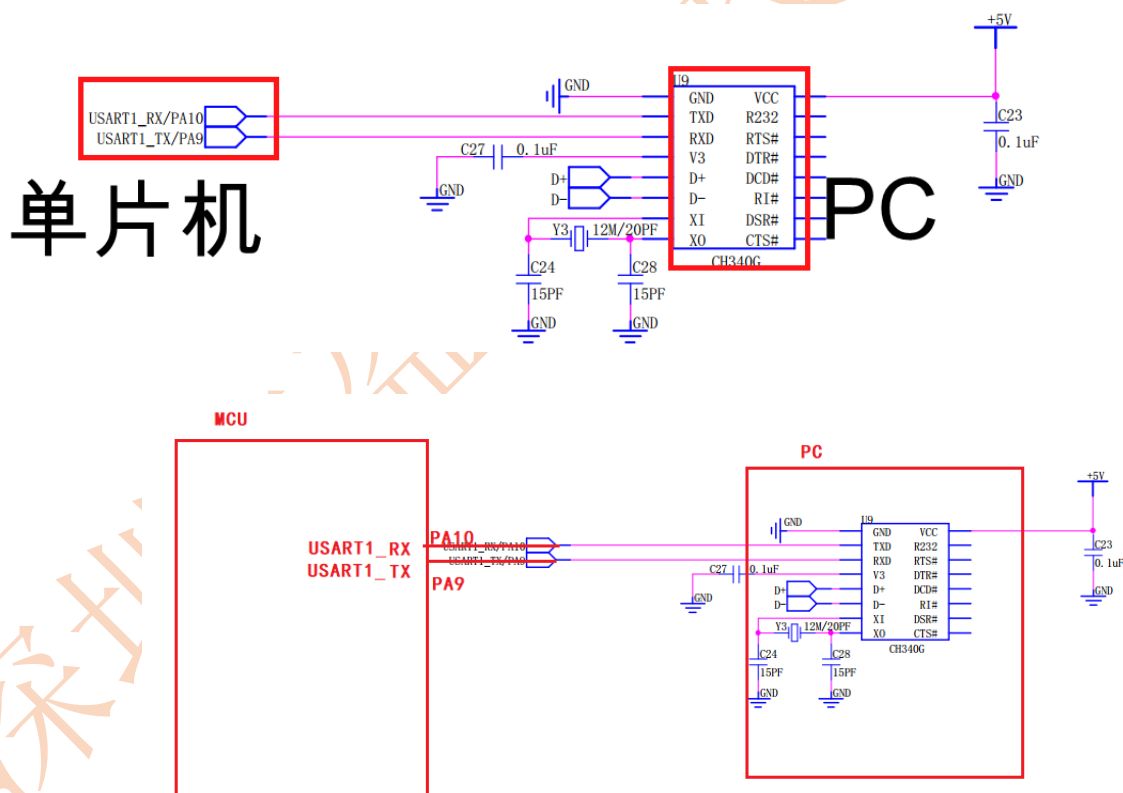
USART1->BRR = MAN_DIV << 4 | FRA_DIV;

3.4 STM32 的 UART 寄存器

26.6.1 状态寄存器 (USART_SR)	标志位	0x08	USART_BRR	
26.6.2 数据寄存器 (USART_DR)	要读写数据		Reset value	
26.6.3 波特率寄存器 (USART_BRR)	存储USARTDIV			
26.6.4 控制寄存器 1 (USART_CR1)				
26.6.5 控制寄存器 2 (USART_CR2)	控制uart功能的寄存器		SART_CR1	

3.5 STM32 的 UART 实验

3.5.1 硬件分析



从上图可得: PA9 可以复用为 USART1_TX

PA10 可以复用为 USART1_RX

所以要将 PA9 和 PA10 配置为复用功能

GPIO 四大功能 MODE

STM32F4xx中文参考... stm32f407vg.pdf * M4-核心板原理图_V2.... PDF万能

Table 9. Alternate function mapping

Port	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	AF9	AF10	AF11	AF12	AF13	AF14	AF15
	SYS	TIM1/2	TIM3/4/5	TIM8/9/10/11	ADC1/2/3	SPB1/SPB2/SPB3	SPB12/SPB13/SPB14	USART1/2/3/4/5/6/7/8	USART4/5/6/7/8	CAN1/2	OTG_FS/OTG_HS	ETH	FSMC/SDIO/OTG_FS	DCMI		
PA0	-	TIM2_CH1_ETR	TIM5_CH1	TIM8_ETR	-	-	-	USART2_CTS	USART4_TX	-	-	ETH_MDCRS	-	-	-	EVENTOUT
PA1	-	TIM2_CH2	TIM5_CH2	-	-	-	-	USART2_RTS	USART4_RX	-	-	ETH_MDI_RX_DV ETH_PMA_REF_CLK	-	-	-	EVENTOUT
PA2	-	TIM2_CH3	TIM5_CH3	TIM8_CH1	-	-	-	USART2_TX	-	-	-	ETH_MDIO	-	-	-	EVENTOUT
PA3	-	TIM2_CH4	TIM5_CH4	TIM8_CH2	-	-	-	USART2_RX	-	-	OTG_HS_ULPL_D0	ETH_MDIO	-	-	-	EVENTOUT
PA4	-	-	-	-	-	SPB1_NSS	SPB12_NSS	USART2_CK	-	-	-	-	OTG_HS_SOF	DCMI_HSYNC	-	EVENTOUT
PA5	-	TIM2_CH1_ETR	-	TIM8_CH1N	-	SPB1_SCK	-	-	-	-	OTG_HS_ULPL_CK	-	-	-	-	EVENTOUT
PA6	-	TIM1_BKIN	TIM3_CH1	TIM8_BKIN	-	SPB1_MOSI	-	-	-	TIM13_CH1	-	-	-	DCMI_PWDK	-	EVENTOUT
PA7	-	TIM1_CH1N	TIM3_CH2	TIM8_CH1N	-	SPB1_MISO	-	-	-	TIM14_CH1	-	ETH_MDI_RX_DV ETH_PMA_REF_CLK	-	-	-	EVENTOUT
PA8	MCIO	TIM1_CH1	-	-	ADC1_SCL	-	-	USART1_CK	-	-	OTG_FS_SOF	-	-	-	-	EVENTOUT
PA9	-	TIM1_CH2	-	-	ADC1_SCK	-	-	USART1_TX	-	-	-	-	-	DCMI_D0	-	EVENTOUT
PA10	-	TIM1_CH3	-	-	-	-	-	USART1_RX	-	-	OTG_FS_ID	-	-	DCMI_D1	-	EVENTOUT
PA11	-	TIM1_CH4	-	-	-	-	-	USART1_CTS	-	CAN1_RX	OTG_FS_DM	-	-	-	-	EVENTOUT
PA12	-	TIM1_ETR	-	-	-	-	-	USART1_RTS	-	CAN1_TX	OTG_FS_DP	-	-	-	-	EVENTOUT

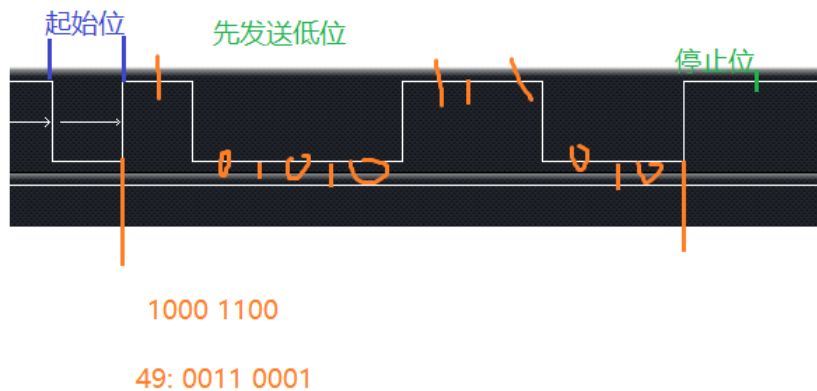
3.5.2 软件设计

配置流程：

1. 先打开 GPIOA 时钟(PA9/PA10)
2. 将 PA9/PA10 配置为复用功能(MODER)
3. 将 PA9/PA10 复用到 USART1(AFRH:AF7)
4. 先打开 USART1 时钟
5. 配置 CR1 寄存器(数据位和奇偶校验)
6. 配置 CR2 寄存器(停止位)
7. 设置波特率(BRR)(求出 USARTDIV 再取出整数和小数 放在相应的位置上)
8. 使能 USART1

围绕着 uart 四要素进行配置





3.6 课后作业

当 MCU 接收到字符串 “LED_ON” 时，板子上的 LED 灯全部点亮。

当 MCU 接收到字符串 “LED_OFF” 时，板子上的 LED 灯全部熄灭。

串口字符串接收：

我们单片机在接收外部设备发送过来的字符串时，根本不知道有多少个，也不知道对方什么时候传输完成。

解决方案：

通信双方规定好以特殊字符作为结束标志

电脑：LED_ON# dsfsd#

超时接收(先不讲 等我们学完定时器来讲)