

Body Area Network

Demo Test Report

haobo.gao

April 22, 2019

Foxconn ZZDC

介绍

我们 BAN 的原本设计是这样的：

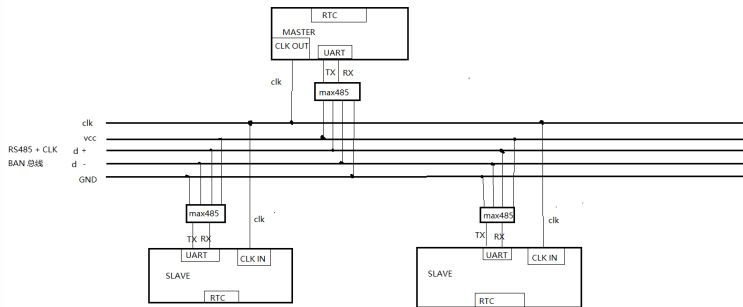


Figure: BAN

1. 由 RS485 增添一根 clk 线形成 BAN 总线的基本硬件线路。
2. clk 用于产生 ms 级别的脉冲波，用于时钟的同步。
3. d+ d- 是差分处理后的数据信号。

图2 假设了挂载在总线上的设备有一个主机，两个从机。



max485 是一型 RS232 转 RS485 的芯片，可以把串口数据转化为差分信号，使其符合 **RS-485** 电气规范。

STM32 方面需要实现串口的驱动，使确保串口可以实时的发送数据。接收数据。



本地时钟用于本地计数，同步时钟用于同步 BAN 上设备的时钟。

- RTC，本地时钟。每个设备都有个本地时钟。计时粒度为 100us.
- CLK, 同步时钟。BAN 总线上的设备使用时钟线来完成时钟同步。主设备的定时器产生同步脉冲，从设备的定时器捕获同步脉冲完成同步。



在 demo 中，为了克服硬件条件的限制，我们：

- 没有使用 max485 来完成 **TTL 信号转差分信号**。
- 使用另外的一个定时器来模拟 RTC 的功能。
- 只能模拟一个主设备，一个从设备的情况。
- 使用串口

下图是 demo 的连线和结构图：

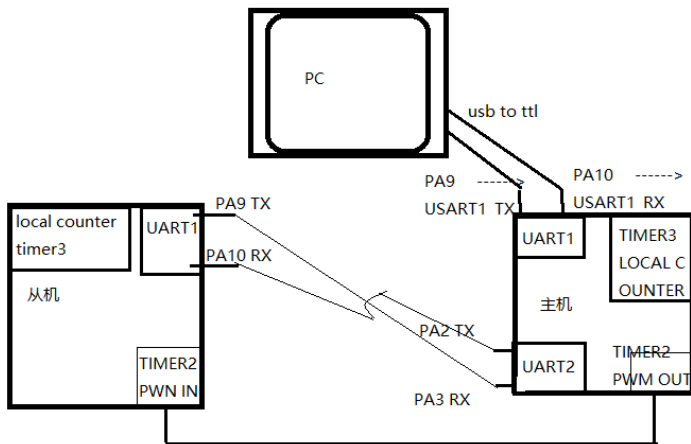


Figure: demo

workflow



1. 主机, 从机上电。外设驱动初始化。从机启 PWM 捕获状态, 等待事件的发生。
2. 主机在接收到 PC 的"begin" 指令的时候。
 - a. 先转发给从机
 - b. 开始 timer3 本地计数 (local counter).
 - c. 开始使用 timer2 产生 PWM 波。
3. 当从机接收到"begin" 指令, 且从机的 timer2 捕获到 clk 上的第一个上升沿。
 - a. timer3 本地计数 (local counter).
 - b. 进入 PWM 捕获同步状态。
4. "report" 指令:
 - a. 主机接收到"report" 指令后, 上报自己的时间戳给 PC。
 - b. 从机接收到"report" 指令后, 上报自己的时间戳给主机, 主机转发给 PC。



local_tm_t 为主从机代码中描述时间的类型:

```
1 typedef struct local_time_struct{
2     unsigned char sec:7;    //秒
3     unsigned char min:7;    //分
4     unsigned char hour:5;   //时
5     unsigned int  day;      //天
6     unsigned int  jiffies;  //本地时间戳
7     unsigned int  jiffies_pwm; //pwm 时间戳
8 } local_tm_t;
```

- 主机使用 timer2 产生的 PWM 波来作为同步信号, 从机捕获 pwm 上升沿。这个方波上升沿间隔固定且精确为 4ms 或者 8ms, 可以客制化, 在方波的上升沿, 会去触发 jiffies_pwm 累加。
- 主从都使用 timer3 每隔 100us 产生一个中断, 在中断中 jiffies 累加, 到 10000 (1s) 超时进位。作为本地时钟。



timer2 的时钟用于产生或捕获 PWM，这个时钟用于同步。

timer3 的时钟仅用于小粒度的时间戳测量。

对于从机来说，每捕获到一个上升沿，为了时间同步，需要强制使从机的 jiffies 归零，jiffies_pwm 增加一个捕获周期（4ms 或者 8ms）。