AN1013
应用笔记

PY32F030/003/002A 系列 USART 的硬件自动波特率检测

前言

正确的 USART 通信要求发送和接收波特率的匹配度足够高，否则可能发生通信错误。当在两个设备之间建立通信链路时，自动波特率检测十分有用，因为从设备能够检测到主控制器的波特率并进行相应的自我调整。这需要使用一种自动机制来确定波特率。PY32F030/003/002A 器件中内置的 USART 外设提供许多功能，包括硬件自动波特率检测。

本应用笔记旨在介绍 PY32F030/003/002A 微控制器的自动波特率检测功能。

在本文档中，PY32 仅指表 1 中列出的产品系列。

表 1. 适用产品

类型	产品系列
微型控制器系列	PY32F030、PY32F003、PY32F002A

目录

1 自动波特率检测模式

1.1 特性概述

自动波特率检测（ ABR）使接收设备能够接受来自各种以不同速率工作的发送设备的数据，无需事先建立数据速率。

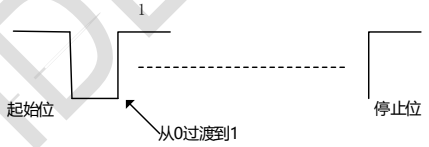
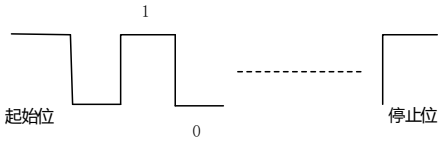
1.2 自动波特率检测模式

ABR 是指接收设备通过检查第一个字符（通常是预先选择的标志字符）确定传入数据速率的过程。

PY32F030/003 系列产品上的自动波特率检测功能内置的各种模式基于不同字符模式：

- 以 bit0 为”1”的任意字符： 模式 0
- 以 bit0bit1 为”10”开头的任意字符： 模式 1

表 1-1 自动波特率检测模式

ABR 模式	说明	波形
模式 0	接收的字符为以 bit0 为“1”的字符。这种情况下， USART 会测量起始位的持续时间（下降沿到上升沿）。	
模式 1	以 bit0bit1 为”10”开头的任意字符。这种情况下， USART 会测量起始位和第一个数据位的持续时间。从下降沿到下降沿测得的持续时间， 可在信号斜率较小时确保较高的精度。	

1.3 ABR 误差计算

由 USART 时钟源（ fCK）决定通信速率范围（尤其是最大通信速率）。接收器采用过采样技术，可区分有效输入数据和噪声，从而用于恢复数据。这可以在最大通信速率与噪声/时钟不准确性之间实现平衡。

USART 时钟源频率必须与预期波特率兼容：

USART 必须选择 16 倍过采样，且波特率在 fCK/65535 和 fCK/16 之间

波特率误差取决于 USART 时钟源、过采样方法和 ABR 模式。

$$误差(\%) = \left| \frac{预期波特率 - 实际波特率}{预期波特率} \right| * 100$$

其中：

- 预期波特率取决于发送设备
- 实际波特率是 USART 接收器使用自动波特率检测操作确定的波特率。

例如：

系统时钟 48M,pclk 配置为 24M,使能 ABR 模式 0。USART 发送器配置波特率为 115200，发送 0x01 给 USART，PY32 USART 通过自动波特率检测功能计算出波特率为(USART_BRR=0xCF) 115942.03。误差约为 0.64%。

2 注意事项

2.1 如何正确接收特率检测字节

用作发起波特率检测的字节（比如模式 0 中以 bit0 为”1”的任意字符，模式 1 中以 bit0bit1 为”10”开头的任意字符）第一次不能被正确接收。这是由于波特率检测误差导致，如果用户想正确接收该字节，按照如下操作流程即可：

操作流程：

- 初始化串口，使能波特率检测
- 通过 Host 发送波特率检测字节给 USART，此时波特已经检测成功
- 设置 USART_SR 寄存器中的自动波特率请求位。该位写 1 会复位 ABRF 标志位，并且请求下一帧的自动波特率检测。
- 再次发送波特率检测字节，USART 会用最新的波特率正确接收该字节

2.2 USART 时钟源频率与波特率之间的关系

USART 时钟源频率必须与预期波特率兼容：

USART 必须选择 16 倍过采样，且波特率在 fCK/65535 和 fCK/16 之间

3 配置示例

3.1 USART1 配置示例

USART1 被配置为自动检测波特率。用户必须在 USART1 初始化函数中选择 ABR 模式，如下所示：

```
/*##-2- Configure the AutoBaudRate method */
UartHandle.AdvancedInit.AdvFeatureInit =UART_ADVFEATURE_AUTOBAUDRATE_INIT;
UartHandle.AdvancedInit.AutoBaudRateEnable =
UART_ADVFEATURE_AUTOBAUDRATE_ENABLE;

/*Uncomment your appropriate mode */

//UartHandle.AdvancedInit.AutoBaudRateMode =
UART_ADVFEATURE_AUTOBAUDRATE_ONSTARTBIT;
//UartHandle.AdvancedInit.AutoBaudRateMode =
UART_ADVFEATURE_AUTOBAUDRATE_ONFALLINGEDGE;

if (HAL_UART_Init(&UartHandle) != HAL_OK)
{
    /* Initialization Error */
    Error_Handler();
}
/* Loop until the end of Autobaudrate phase */
while(__HAL_UART_GET_FLAG(&UartHandle,UART_FLAG_ABRF) == RESET);
```

在整个初始化过程完成后， USART 等待从超级终端接收数据，然后开始自动波特率检测阶段。通过 ABRF 标志监测此阶段的结束。

- 如果自动波特率检测操作不成功，则 ABRE 标志置位
- 如果自动波特率检测操作成功完成，则向超级终端发送确认数据。

```
/* If AutoBaudBate error occurred */
if (__HAL_UART_GET_FLAG(&UartHandle, UART_FLAG_ABRE)!= RESET)
{
    Error_Handler();
}
else
{
    /* Wait until RXNE flag is set */
    while(__HAL_UART_GET_FLAG(&UartHandle,UART_FLAG_RXNE) == RESET);
    /* Send acknowledgement message*/
    if (HAL_UART_Transmit(&UartHandle, (uint8_t *)aTxBuffer, TXBUFFERSIZE,0x1FFFFFF)
      != HAL_OK)
    {
        /* Transfer error in transmission process */
        Error_Handler();
    }

    while (HAL_UART_GetState(&UartHandle) != HAL_UART_STATE_READY)
    {
    }

}
//该位写 1 会复位 ABRF 标志位，并且请求下一帧的自动波特率检测。
SET_BIT(UartHandle.Instance->SR, USART_SR_ABRRQ);
```

4 版本历史

版本	日期	更新记录
V0.1	2021.10.15	初版
V1.0	2022.7.4	修改格式
V1.0	2022.10.24	增加 002A 内容

Puya Semiconductor Co., Ltd.