

DOI:10.15923/j.cnki.cn22-1382/t.2017.6.04

# 标准 51 单片机在任意时钟频率下 产生任意波特率的独特方法

林志琦, 任超越\*, 张睿

(长春工业大学 电气与电子工程学院, 吉林 长春 130012)

**摘要:** 为了实现标准 51 单片机在 6、12、24 MHz 等时钟频率下产生任意的波特率, 将波特率发生器中定时计数器的 16 次(或 32 次)溢出分为两部分, 一部分是连续 15 次(或 31 次)的快速溢出, 此时定时器  $T_1$  的初值为 0FFH, 定时器只需一个定时周期就可以溢出, 这样需要 15 个(或 31 个)时间单位, 然后用串行数据传输一位数据所需的时间减去之前那 15(或 31 个)定时周期, 余下的时间为用定时器  $T_0$  准确地定时。故此可以在任意时钟频率下产生任意的波特率。

**关键词:** 51 单片机; 时钟频率; 波特率; 波特率发生器

**中图分类号:** TN 209; TP 368.1 **文献标志码:** A **文章编号:** 1674-1374(2017)06-0537-07

## A special method for 51 MCU to generate arbitrary baud rate at any clock frequency

LIN Zhiqi, REN Chaoyue\*, ZHANG Rui

(School of Electrical & Electronic Engineering, Changchun University of Technology, Changchun 130012, China)

**Abstract:** To generate an arbitrary baud rate at 6 MHz, 12 MHz and 24 MHz clock frequency for 51 MCU, the overflow of 16 (or 32) times of timers in the baud rate generator are divided into two parts. The one is the rapid overflow of 15 consecutive (or 31) consecutive times. The initial value of the timer  $T_1$  is 0FFH, and only one regular cycle is needed to generate the overflow which requires 15 (or 31) time units. The time needed to serial transport one byte data minus the 15 consecutive (or 31) consecutive time, and the remain time is the one for timer  $T_0$ . Therefore, arbitrary baud rate can be generated at any clock frequency.

**Key words:** 51 single chip microcomputer; clock frequency; baud rate; automatically reloaded.

收稿日期: 2017-09-14

基金项目: 长春市重大科技攻关计划项目(13KG09)

作者简介: 林志琦(1964—), 男, 汉族, 吉林长春人, 长春工业大学教授, 博士, 主要从事电气及自动控制方向研究, E-mail: 550979759@qq.com. \* 通讯作者: 任超越(1992—), 男, 汉族, 内蒙古赤峰人, 长春工业大学硕士研究生, 主要从事电力电子与电力传动方向研究, E-mail: 765326224@qq.com.

### 0 引言

随着计算机网络化与微机分级分布式应用系统的发展,通信在各方面显得尤为重要。通信是指计算机与外界进行信息传输,既包含计算机和计算机之间的传输,也包括计算机和外部设备,例如终端、扫描仪和移动硬盘等设备之间的传输<sup>[1]</sup>。单片机的串口通信是自动化工业、智能终端、光电系统等领域非常重要的信息传输手段。在光电系统中,单片机一般充当控制器的角色,通过串口发送一定格式的数据来控制与之相连设备的动作,同时设备也会将一些自身的状态信息反馈给单片机,供单片机进行判断并进行相应的控制<sup>[2]</sup>。例如对云台的控制,当键盘输入信息时,单片机和软件相互配合对信息进行识别,根据所识别的信息向云台中的步进电机发出指令,实现对方向角、俯仰角控制、速度控制和程序控制等功能<sup>[3]</sup>。云台控制器通常以单片机为核心,通过单片机串口通信实现控制。

文中对标准 51 单片机串口数据传输的方式进行了改进。标准 51 单片机在进行串口传输数据时存在不足,当串口传输速度增加时,波特率误

差将会增大,随着误差的增大,传输的数据会出现错误,直接影响了串口传输的质量。文中经过对标准 51 单片机传输方式的改进,串口波特率误差明显减小,串口传输质量显著提高。

### 1 51 单片机的结构及传输方式

标准的 51 单片机有 2 个 16 位的定时计数器,2 个外部中断,5 个中断源,1 个全双工的异步串行通信接口<sup>[4]</sup>。在标准的 51 单片机中,没有专门的波特率发生器,而是由定时计数器 T<sub>1</sub> 的溢出作为串口数据传输的定时基准。在标准 51 单片机刚推出时,串口的数据传输速率在 300 ~ 2 400 bit/s,传输每位二进制数的时间在 416 ~ 3 300 μs 之间,而当时的 51 单片机指令周期为 1 μs,采用 8 位自动加载的工作方式时,最大溢出时间为 256 μs,不能达到传输一位数据所需的时间。为了满足 416 ~ 3 300 μs 的定时时间,标准 51 单片机的设计者在定时 T<sub>1</sub> 的溢出与串行数据收发移位寄存器之间加入了一个 16/32 进制可选的计数器,对定时计数器的溢出进行分频,使其定时时间得到延长,从而满足 416 ~ 3 300 μs 的定时需求<sup>[5]</sup>,其工作原理如图 1 所示。

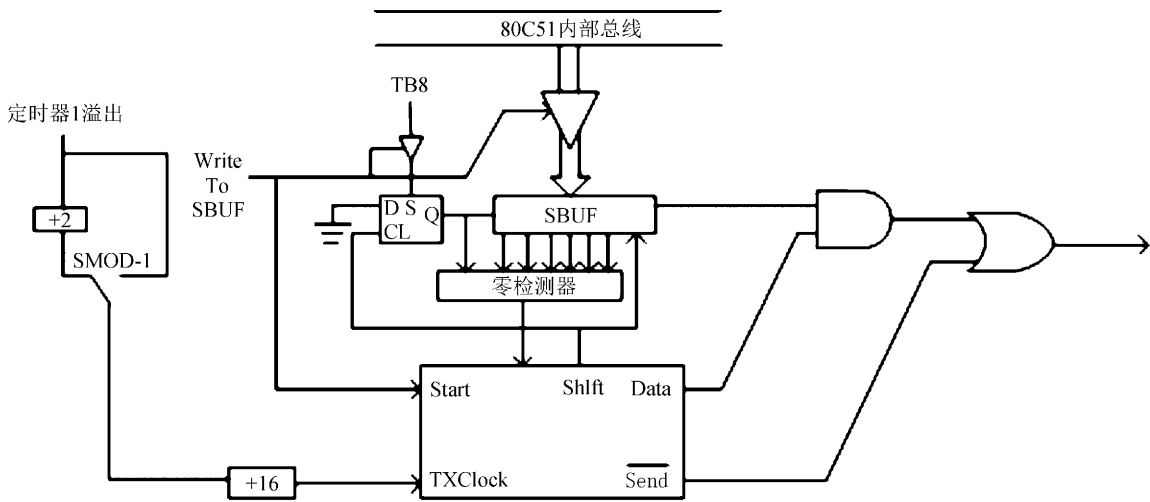


图 1 51 单片机波特率发生器工作原理图

### 2 标准 51 单片机串口传输的误差分析

标准 51 单片机每计一个数的时间为一个机器周期,一个机器周期等于 12 个时钟周期,如果用 12 MHz 晶振时产生 9 600 bit/s 的波特率,其误差为 3.3%,它是指传输一位数据的误差,当传

输一帧数据时的误差为 33%,随着误差的增大,传输的数据会出现错误,其误差过大的原因如下:

采用波特率计算公式:

$$\text{波特率} = \left( \frac{2^{\text{SMOD}}}{32} \right) \times (\text{T}_1 \text{ 溢出率}) \quad (1)$$

$$\text{溢出率} = \frac{\text{晶振频率}}{256 - X} \times 12 \quad (2)$$

式中:  $X$  —— 定时器初值。

这里以晶振频率 12 MHz, SMOD=0, 波特率=9 600 bit/s 为例, 根据式(1)、式(2)推导出:

$$X = 256 - \frac{\text{晶振频率}}{\text{波特率} \times 32 \times 12} \quad (3)$$

由式(3)可以看出, 如果定时器初值  $X$  想得到一个整数值, 那么晶振频率必须是波特率乘以 32 再乘以 12 的整数倍, 通过计算可知 6、12、24 MHz 等晶振频率都不是波特率乘以 32 再乘以 12 的整数倍, 所以  $X$  不可能为整数值, 然而在装载定时器初值时,  $X$  必须为整数(因为单片机计时时只能记录整数个时钟周期), 所以产生误差, 其误差产生的来源是由于  $T_1$  溢出率存在误差。选择 9 600 bit/s 的波特率, 传送一个位数据所需要的时间为 104.167  $\mu$ s, 那么

$$\text{传送一位数据的时间} = \frac{32}{2^{\text{SMOD}} \times T_1 \text{ 溢出率}} \quad (4)$$

其中,  $T_1$  溢出率在 6、12、24 MHz 等晶振频率下(见图 1), 经过 16 分频和 2 分频, 在 SMOD=0 时, 根据式(4)  $T_1$  的误差放大了 32 倍, 造成了误差大于 3.3% 无法发送准确数据的情况, 所以 51 单片机在 6、12、24 MHz 等晶振频率下无法产生任意波特率。

由于标准的 51 单片机的这种结构, 当串口通信速度由 300 ~ 2 400 bit/s 提高到 9 600 ~ 115 200 bit/s, 甚至 256 000 bit/s 时, 就会出现误差大于 3.3% 的情况, 这也是标准 51 单片机设计的一个缺陷。以 9 600 bit/s 的波特率为例进行分析。9 600 bit/s 的串行数据传输速率, 传输每位二进制数据所需的时间为  $1/9\ 600 = 104.167 \mu\text{s}$ , 这时要求  $T_1$  的溢出时间为  $104.167 \mu\text{s}/32 = 3.255 \mu\text{s}$  或  $104.167 \mu\text{s}/16 = 6.510 \mu\text{s}$ , 在通常的 12 MHz 时钟频率下, 计数时间单位是 1  $\mu\text{s}$ , 要产生 3.255  $\mu\text{s}$  或 6.510  $\mu\text{s}$  的延时, 只能用 3  $\mu\text{s}$  或 4  $\mu\text{s}$  近似, 这时的误差为

$$\frac{3 \times 32 - 104.2}{104} \times 100\% = -7.7\%$$

和

$$\frac{4 \times 32 - 104.2}{104} \times 100\% = 23\%$$

这样采取舍入的方法误差见表 1。

由表 1 可见, 误差已超过了最大误差 3.3%。

标准 51 单片机发送一帧数据时, 每 3 位检测一次, 以每个数据帧 10 位数据, 能识别的最大误差为 33% 计算, 则每个数据位的误差应小于  $(33\%)/10 = 3.3\%$ 。

表 1 在时钟频率为 12 MHz、波特率为 9 600 bit/s 的情况下传输每位二进制数的时间误差

定时器设置及误差	分频系数			
	16		32	
计算的延时时间	6.51 $\mu$ s		3.25 $\mu$ s	
舍去后的延时时间	6 $\mu$ s	7 $\mu$ s	3 $\mu$ s	4 $\mu$ s
定时器 $T_1$ 的初值	256-6	256-7	256-3	256-4
波特率误差	-7.7%	7.7%	-7.7%	23%

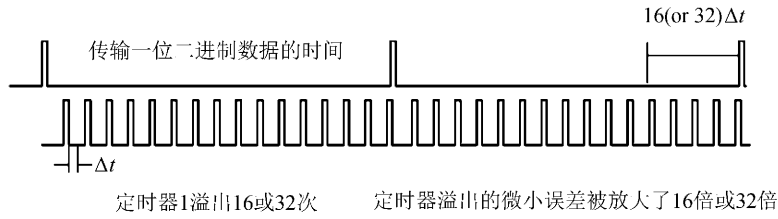
通过上面的分析知道, 在波特率增加后, 每位二进制数据位的传输时间变短, 而这个时间的 16 或 32 分之一就更小, 而这时定时计数器以 1  $\mu$ s 为时间单位的情况下, 小数点后面的舍取对定时时间的影响很大, 即这个误差被放大了 16 倍或 32 倍<sup>[6]</sup>。定时时间大一个或小一个时间单位, 它们的误差都超过了 3.3%。也就是由于标准 51 单片机波特率产生方式的特殊结构, 在 6、12、24 MHz 等时频都超过了 3.3%。也就是说, 由于标准 51 单片机波特率产生方式的特殊结构, 在 6、12、24 MHz 等时频率下产生不了任意的波特率。

在实际工作中, 通常采用可行的方法是选择一个小数时钟频率, 如 11.059 2 MHz, 或 22.118 4 MHz, 此时可以产生 9 600, 19 200 和 38 400 的波特率, 但很难产生 57 600 和 115 200 等其他波特率。同时由于采用非整数时钟频率, 使得定时很难产生整数的定时时间。而在标准的 51 单片机中为了产生整数的定时时间, 通常采用 6、12、24 MHz 等时钟频率, 为了解决这一矛盾, 文中提出了一种可以让标准 51 单片机在 6、12、24 MHz 等时钟频率下产生任意波特率的独特、新颖的方法。

### 3 任意时钟频率下产生任意的波特率的方法

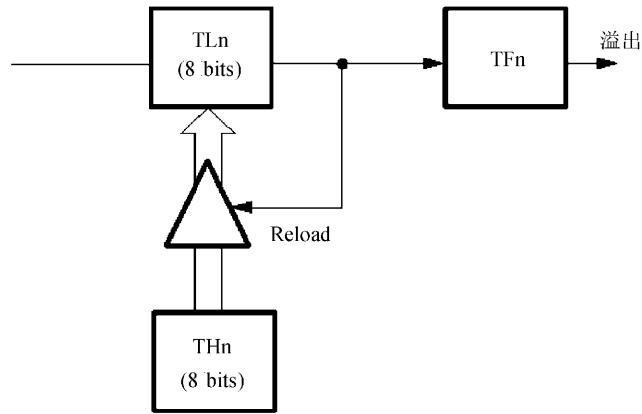
由以上分析可知, 产生波特率误差的原因由于对定时器  $T_1$  的溢出时间进行了 1/16 或 1/32 的分频, 使得定时器  $T_1$  的溢出时间变短并且带有小数, 同时误差也被放大了, 实现这种去掉误差

放大现象的方法如图 2 所示。



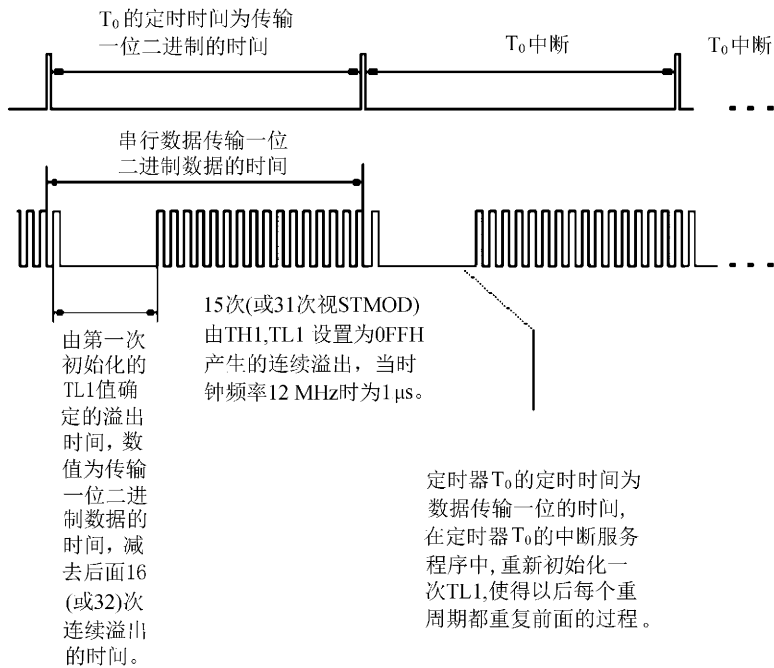
(a) 累积误差

第一次初始化为传输一位所需的时间减去 16 个或 32 个计数时间单位(视 SMOD 而定)



初始化为 0FFH 产生 15 个或 31 个连续的溢出

(b) 特殊方法



定时器  $T_0$  的定时时间为数据传输一位的时间, 在定时器  $T_0$  的中断服务程序中, 重新初始化一次  $TL1$ , 使得以后每个重周期都重复前面的过程。

(c) 自动重新加载

图 2 去掉误差放大现象的方法

标准 51 单片机的波特率累计误差信号的产生见图 2(a), 从图中可以看到, 收发一位串行数据的时间被平均分为 16 份或 32 份, 这样对每份的精度要求就提高了很多, 如果采取特殊的方法, 不是平均分配每次溢出的时间, 那么就可以减小对每次溢出的精度要求, 基于这种想法, 我们将 16 次(或 32 次)溢出分为两部分, 一部分是连续的快速 15 次(或 31 次)溢出<sup>[7]</sup>, 此时定时器 T<sub>1</sub> 的初值为 0FFH, 定时器只需一个定时脉冲就可以溢出, 这样需要 15 个(或 31 个)时间单位, 另外一部分定时时间为串行数据传输一位所需的时间减去那 15(或 31 个)个单位时间, 此时上电初始化的 TL1 的值由这个时间确定:

$$TL1 = 256 - \left[ \frac{\frac{10^6}{burd}}{\frac{12}{f_{osc}}} - \frac{12}{f_{osc}} \times 15(\text{or}31) \right] = 256 - \frac{10^6 \times f_{osc}}{12 \times burd} + \frac{12 \times 15(\text{or}31)}{f_{osc}} \quad (5)$$

标准 51 单片机的定时器工作方式为自动重

新加载方式<sup>[8]</sup>, 开始定时由 TL1 中的初值开始进行加计数, 加到 0FFH 后再加一溢出, 同时把其他 TH1 中的数值 0FFH 自动加载到 TL1 中, 继续进行第二次定时, 在上电初始化时 TL1 满足传输一位二进制数减去 15(或 31)的定时时间值, 这样就可以高精度的实现总的溢出时间为传输一位二进制数时间的波特率(发生器见图 2(b))。当然还有一个最大的问题, 这种方法只能产生一次, 满足要求的波特率发生器定时, 因为除了第一次的定时时间为初始化时的 TL1 值外, 以后就都是由重新加载值 0FFH 了, 为了重复以上过程, 不断产生满足要求的波特率信号, 在连续送 15(31)次连续快速的溢出后要重新初始化一次 TL1, 这个任务由定时器 T<sub>0</sub> 完成<sup>[9]</sup>。

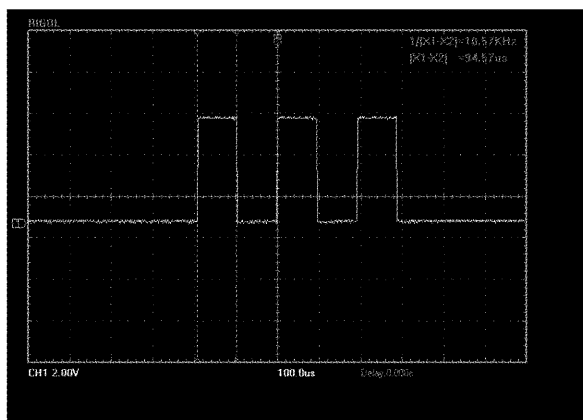
将 T<sub>0</sub> 定时时间为传输一帧数据的时间<sup>[10]</sup>。在 T<sub>0</sub> 中断服务程序中, 对 TL1 重新初始化一次, TH1 仍然保持 0FFH 不变, 就可以使上面的高精度波特率发生过程不断重复下去(见图 2(c))。

经过在多种型号的标准 51 单片机上实验验证时得到的实验数据见表 2。

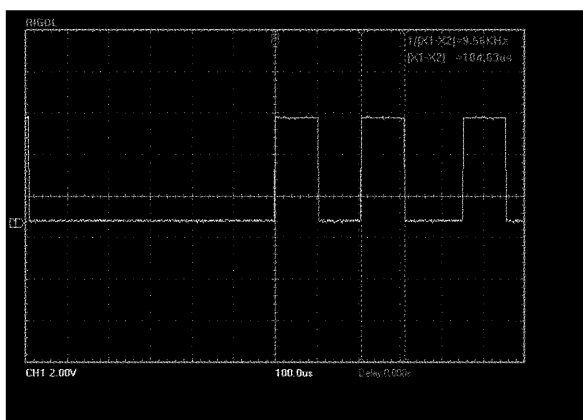
表 2 标准 51 单片机在 6、12、24 MHz 的时钟频率下产生任意的波特率的初始值

fosc		6 MHz		12 MHz		24 MHz	
		TL1	TL0, TH0	TL1	TL0, TH0	TL1	TL0, TH0
9 600	SMOD=1	219	204	166	151	—	—
	SMOD=0	235	204	—	—	—	—
14 400	SMOD=1	221	236	200	185	—	—
	SMOD=0	221	253	—	—	—	—
19 200	SMOD=1	230	245	219	204	165	151
	SMOD=0	—	—	235	204	—	—
28 800	SMOD=1	239	254	221	236	186	170
	SMOD=0	—	—	221	253	—	—
38 400	SMOD=1	—	—	230	245	—	—
	SMOD=0	—	—	—	—	235	204
57 600	SMOD=1	—	—	239	254	—	—
	SMOD=0	—	—	—	—	221	253
115 200	SMOD=1	—	—	—	—	239	254
	SMOD=0	—	—	—	—	—	—

根据实验,传输一位数据的时间波形如图 3 所示。



(a) 改善之前



(b) 改善之后

图 3 传输一位数据的时间波形

没有改进之前的波形见图 3(a),传输一位数据所需要的时间为  $94.57 \mu\text{s}$ ,而理论上波特率为 9 600 时,所需要的时间为  $104.167 \mu\text{s}$ ,此时传输每位二进制数的误差为  $-9.21\%$ 。不能准确收发数据,改进之后传输数据波形见图 3(b)。传输一位数据所需要的时间为  $104.63 \mu\text{s}$ ,此时传输每位二进制数据的误差为  $0.44\%$ ;数据收发无误。

标准 51 单片机在 6、12、24 MHz 的时钟频率下产生任意的波特率的测试程序:

```
#include <REG51.H> /* special function register
declarations */
void main (void)
{
    PCON = 0x80;
    SCON = 0x50; /* SCON: mode 1, 8 - bit
UART, enable rcvr */
```

```
TMOD |= 0x22; /* TMOD: timer 1, timer
0, mode 2, 8bit reload */
TH1 = 0xff; /* TH1: reload value for
1200 baud @ 16MHz */
TL1 = 237; /* show tab 1 */
TH0 = 231; /* about baud show tab 1
*/
TL0 = 231; /* show tab 1 */
TCON = 0x50; /* timer 1 & timer 0 run
while one time */
PT0 = 1;
ET0 = 1;
EA = 1;
while (1) { /* test program loop output to uart
0~255 */
    char i;
    for(i=0;i<255;i++)
    {
        SBUF=i;
        while (! TD)
        {
            TI=0;
        }
    }
}
void plus_int1(void) interrupt 1
{
    TL1 = 237; /* show tab 1 */
}
```

## 4 结 语

根据标准 51 单片机串口传输的缺陷,提出了文中特殊改进方法,并介绍了这种特殊方法的工作原理。实验结果证明,对标准 51 单片机串口传输的方式改进后,串口传输一位的误差为  $0.44\%$ ,相比改进之前的误差有了明显的改善。实现了标准 51 单片机在 6、12、24 MHz 等时钟频率下能够产生任意的波特率。

## 参考文献:

- [1] 李鹏. MCS-51 系列单片机串行通信波特率的研究[J]. 电脑知识与技术, 2013(28): 6265-6267.
- [2] 宣明. 微型机械及相关理论和技术[J]. 光学精密工程, 1994, 2(3): 1-5.
- [3] 韩双喜. 基于 DSP 的运动姿态控制系统设计[J]. 世界电子元器件, 2013(2): 44-47.

- [4] 陈粤初, 窦振中. 单片机应用系统设计与实践[M]. 北京: 北京航空航天大学出版社, 1991.
- [5] 李会杰, 李俊廷, 张伟, 等. 一种51系列单片机的解密方法[J]. 长春工业大学学报, 2017, 38(4): 335-339.
- [6] 于建群, 王立鼎, 刘军山. 集成毛细血管电泳芯片的结构及其制作技术研究进展[J]. 中国机械工程, 2003, 14(2): 166-175.
- [7] 徐骏平, 胡兵, 马平化, 等. USB接口技术在激光打标中的应用[J]. 激光杂志, 2006, 27(3): 77-78.
- [8] 陈翠华, 柴金华. 模数转换与波特率设置对大气激光语音通信质量的影响[J]. 量子电子学报, 2006, 32(2): 225-230.
- [9] 林志琦, 李会杰, 郎永辉, 等. 用四象限光电探测器获得光斑参数[J]. 光学精密工程, 2009, 17(4): 764-770.
- [10] 张新宝, 王海璇, 周胜利, 等. 基于串口提高MCU-51矩形单脉冲脉宽精度的方法[J]. 吉林大学学报, 2016, 34(4): 496-500.