

TM57ML40

RFC 电阻测量应用范例

For C Language

Application Note

tenx reserves the right to change or discontinue the manual and online documentation to this product herein to improve reliability, function or design without further notice. tenx does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. tenx products are not designed,

intended, or authorized for use in life support appliances, devices, or systems. If Buyer purchases or uses tenx products for any such unintended or unauthorized application, Buyer shall indemnify and hold tenx and its officers, employees, subsidiaries, affiliates and distributors harmless against all claims, cost, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use even if such claim alleges that tenx was negligent regarding the design or manufacture of the part.

AMENDMENT HISTORY

| Version | Date | Description |
|----------------|-------------|--------------------|
| V1.0 | Jan, 2012 | New release |

CONTENTS

AMENDMENT HISTORY.....2

概要：4

1. 具体 DEMO 程序请参考"TM57ML40_RFC.C"4

2. 范例说明4

3. 线路图.....8

TM57ML40_RFC 电阻测量应用范例

概要:

本文以 **TM57ML40** 为例,来演示使用 **RFC** 测试电阻,并通过 **LED** 显示出所测试的值

1. 具体 DEMO 程序请参考"TM57ML40_RFC.C"

2. 范例说明

2-1 RFC 由 RC 振荡器与 RFC 计数器组成, RFC 包含 6 个管脚 CX0-1 为施密特触发输入震荡, RFC0-3 电阻输出引脚。

2-2 TM57ML40 有 3 种 RFC 控制模式

- 1) 正常模式: 由 RFCSTOP 位控制 RFC 的开启和关闭。
- 2) TM0 模式: TM0STPRFC 被置 1, RFC 计数器由 TM0TGI 控制。
- 3) TM1 模式: TM1STPRFC 被置 1, RFC 计数器由 TM1TGI 控制。

2-3 在程序例程中使用的是 TM0 模式, 并且驱动 4 位 LED 来显示所测试的数据, 这里 LED 驱动的方法就不在说明了, LED 的驱动方法可参 LED_LCD.ASM 程序。RFC 的工作设置步骤如下:

- 1) 将所使用 RFC 的 I/O 端口数据寄存器设置为 1。
PBD=0XFF //is input
- 2) 将所使用 RFC 的输入输出寄存器设置为 0, 既输入。
PBE=X0E3 //is input
- 3) 关闭所使用 RFC 端口的上拉寄存器。
NPBPU=0XFF

2-4 设置定时器 0 的计数值与分频, 程序中的计数值是 131 分频是 64, 定时时间为 4 ms。要想 RFC 测试出来的精度越高的话, 首先需要保证电压的稳定以及参考电阻的精度, 再就是采样的时间延长, 理论上时间越长越好, 但是也需要根据程序方便计算的来设置采样时间。

```
TIMER0=131           //Timer 0 data
TM0RELOAD=131        //Time0 Reload Data
OPTION=0X06           //bit0-3 定时器分频
```

2-5 打开定时器 0。

```
TM0IE=1              //打开定时器 0
```

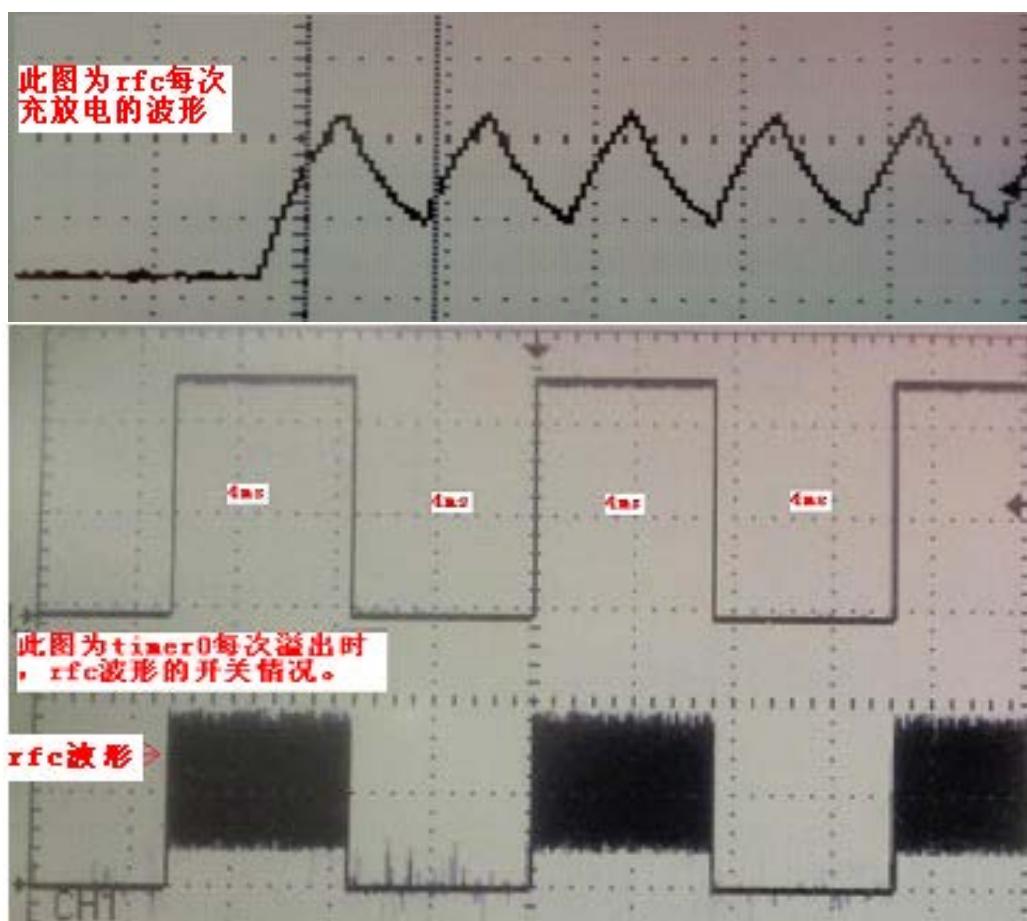
2-6 设置 RFC 由定时器 0 中断溢出控制, 选择 CX0 为 RFC 计数时钟, 选择 RFC0 pin 为 RFC 的波形产生脚。

```
RFCCON=0x41          //bit6 Timer0 overflow to control RFC
                      //bit4 select CX0 as RFC Counter CLK
                      //bit0 select RFC0 pin for RFC loop
```

2-7 打开 RFC 中断, 清零 RFC 计数器以及打开 RFC 工作位。

| | |
|-----------|--------------------------|
| RFCINTE=1 | //RFC Interrupt Enable. |
| CLRRFC=0 | //RFC 计数器清零 |
| RFCSTOP=0 | //start RFC Counter/Loop |

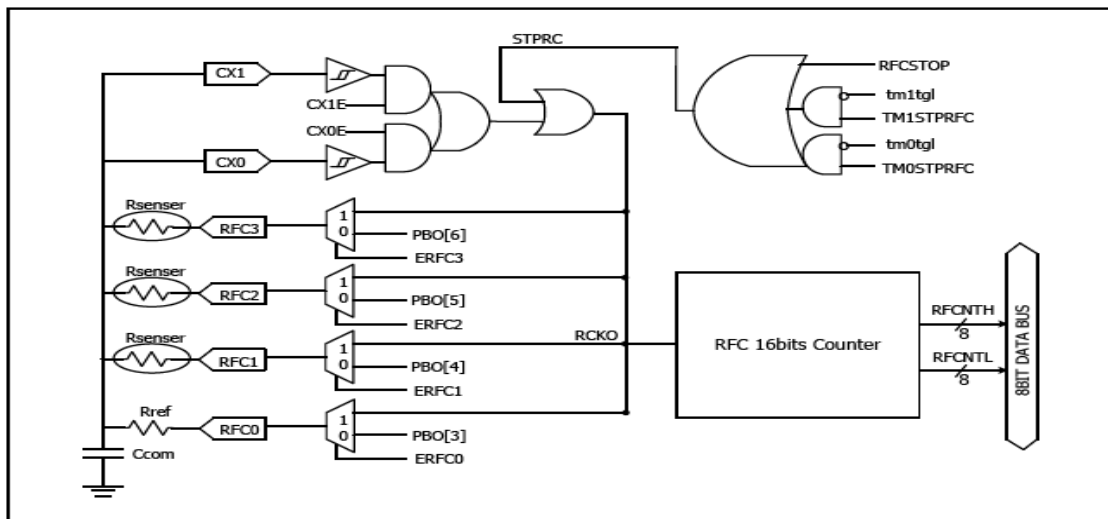
2-8 依据上述步骤设置完后 RFC 能够正常的从 RFC0 pin 输出波形，RFC 电路中使用的是 10K 电阻和 102 电容，RFC0 作为参考电阻测试引脚，当定时器 0 溢出中断时，RFC 根据之前的状态来判断是产生波形还是停止产生波形，RFC 的波形产生总是根据定时器 0 的溢出循环产生的。如下图所示：



2-9 当 RFC 有产生上图所示波形时，就说明 RFC 设置时没有问题的，下面就了解下 RFC 的结构，RFC 的结构如下图所示。

针对下图，分析下三个工作模式，

- 1) 正常模式：由 RFCSTOP 位控制 RFC 的开启和关闭，在使用 RFCSTOP 位控制时必须关闭 TM0STPRFC 和 TM1STPRFC，在 RFCSTOP=0 时，RFC 波形一直产生是，直到 RFCSTOP=1 时停止。
- 2) TM0 模式：TM0STPRFC 被置 1，RFC 计数器由 TM0TGI 控制，但是 RFCSTOP 位必须等于 0，此时 RFC 波形的产生由 timer0 的溢出控制 TM0TGI 的翻转。
- 3) TM1 模式：TM1STPRFC 被置 1，RFC 计数器由 TM1TGI 控制，但是 RFCSTOP 位必须等于 0，此时 RFC 波形的产生由 timer1 的溢出控制 TM1TGI 的翻转。



2-10 了解了以上的内容就可以写程序了，具体程序参见请 `rfc.C`，这里列出计算公式以及程序流程图供大家参考。

定时的基本时间/参考电阻RFC值=参考电阻单个RFC时间

定时的基本时间/被测电阻RFC值=被测单个RFC时间

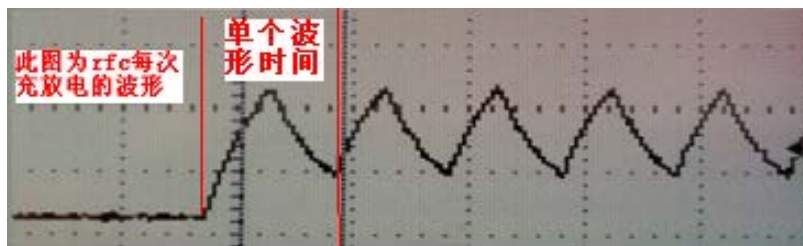
被测电阻单个RFC时间*参考电阻阻值/参考电阻单个RFC时间=被测电阻阻值

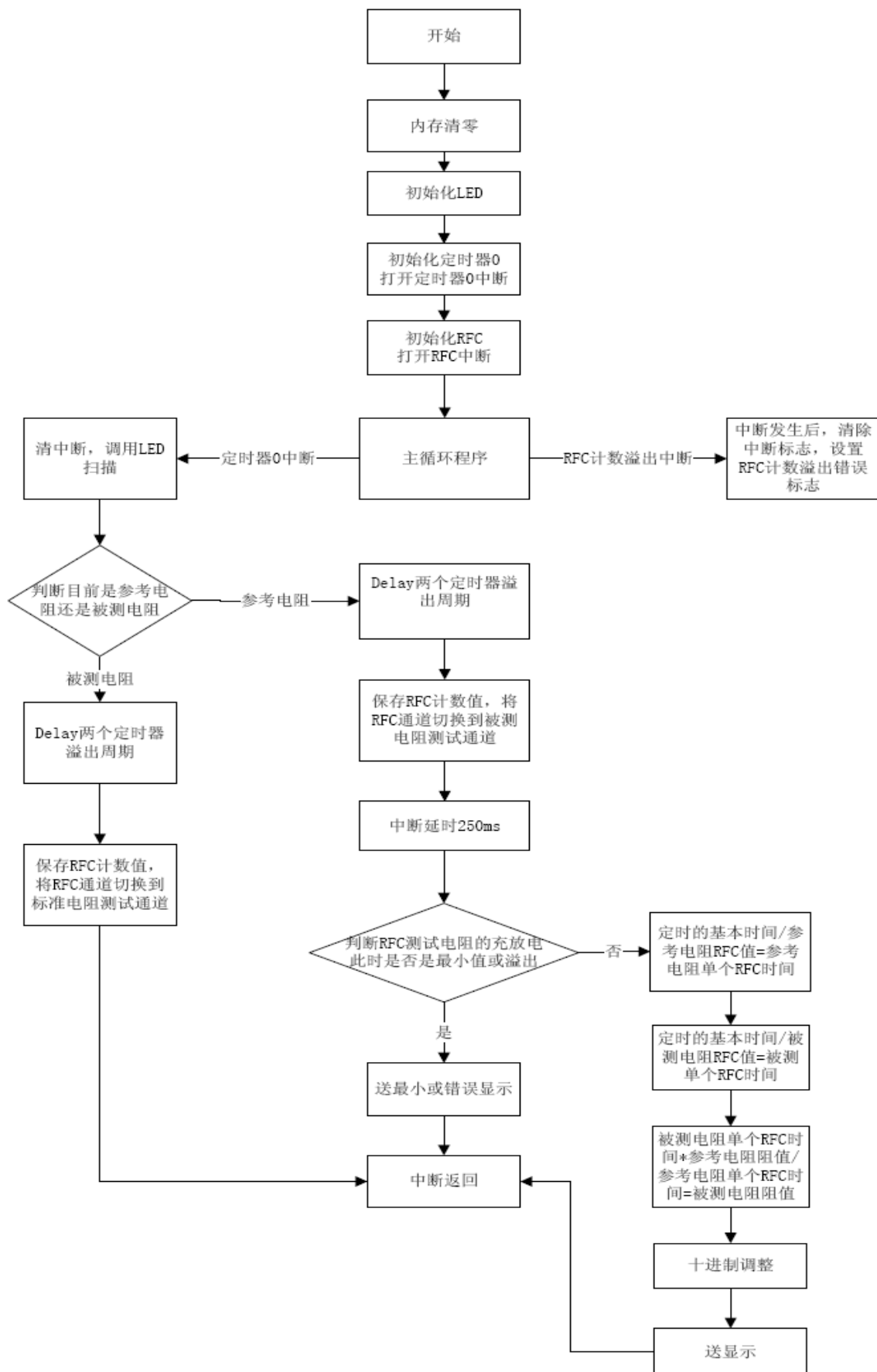
定时的基本时间：即定时器0的定时时间，程序中位4 ms=4000 us。

参考电阻RFC值：即RFC0通道时测试的时间程序中保存在`rfc_count_h`与`rfc_count_l`中。

被测电阻RFC值：即RFC1通道时测试的时间程序中保存在`rfc_count2_h`与`rfc_count2_l`中。

参考电阻阻值：既与被测电阻比较的电阻，demo中选用的是10K。





3. 线路图

