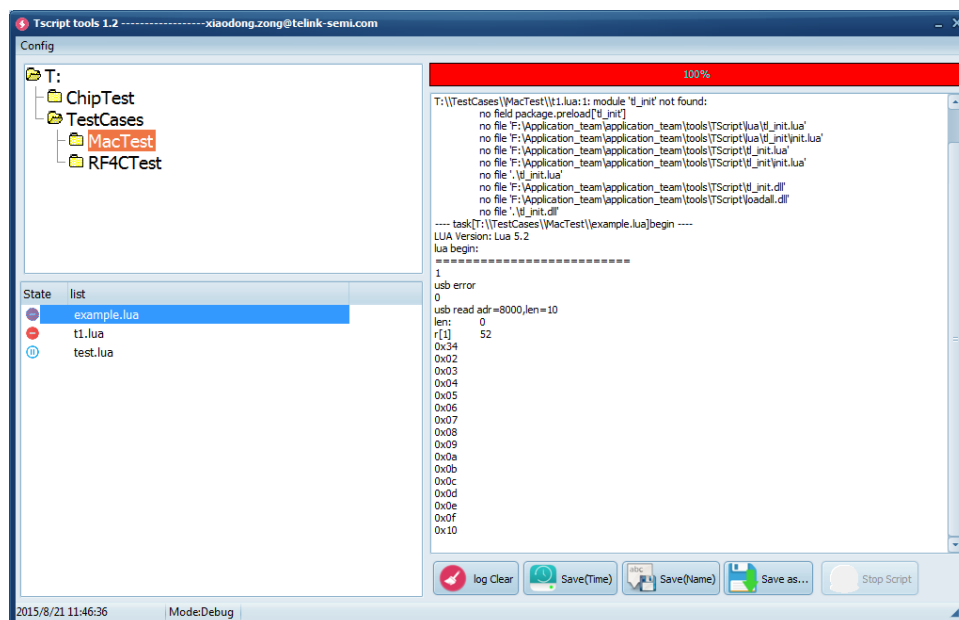


# Tscript 使用说明

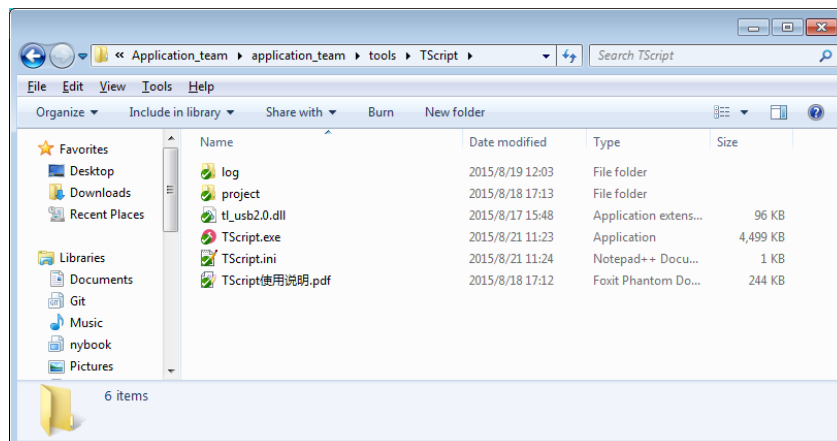
2015/8/18 宗晓东

## 1 工具介绍



Tscript 工具主要用来运行 lua 脚本, 在 lua 脚本中可以执行一些 USB 的操作函数.从而可以做一些芯片测试.

工具的目录结构如下:



project 目录: 放置 lua 脚本的目录,里面可以建立任意子目录,脚本扩展名为\*.lua

log 目录: 默认 log 的保存目录,点击 Save(Time)按钮可以保存 log 窗口的内容到 log 目录下, 文件名类似: log2015-8-19 11h22m27s.txt, 点击 Save(Name)按钮,保存 log 窗口的内容到 log 文件夹下,文件名为脚本名字类似: Case\_t1.txt.

Tscript.ini 配置文件, 配置 Tscript.exe 的一些基本属性, 目前只有两个:

EDITOR= "C:\Windows\notepad.exe" : 默认编辑器  
DEBUGMODE = 0 : 默认调试模式

## 2 脚本介绍

Tscript.exe 中使用的脚本为 lua 格式,文件扩展名为\*.lua

其中加入了几个自定义函数:

### 1 **int tl\_usb\_init(id)**

usb 初始化函数

id : 芯片的{0x7f,0x7e}地址里的芯片 ID

返回值为 usb 的设备句柄

例子:

```
handle = tl_usb_init(0x1234)
```

### 2 **int tl\_usb\_write(handle, adr, buffer, len)**

handle : 通过 tl\_usb\_init 获得的 usb 设备句柄

adr : 地址

buffer : array 类型的变量

len : 写数据的长度

返回值为写成功的数据长度

例子:

```
a = array.new(4)
a[1] = 0x11; a[2] = 0x22; a[3] = 0x33; a[4] = 0x44;
result_len = tl_usb_write(handle,0x8000,a,16)
```

### 3 **r,r\_len = tl\_usb\_read(handle, adr, len)**

handle : 通过 tl\_usb\_init 获得的 usb 设备句柄

adr : 地址

len : 写数据的长度

返回值 r 为 table 类型的变量, 保存读到的数据值

返回值 r\_len 为读取到的数据长度

例子:

```
r,r2 = tl_usb_read(handle,0x8000,16)
```

### 4 **tl\_sleep\_ms(t)**

睡眠函数

t : 睡眠时间(ms 为单位)

### 5 **tl\_error(error\_bit)**

报错函数

error\_bit: 错误标志

### 6 **tl\_progress(pos)**

进度条控制函数

pos : 设置进度条的当前值(0 ~ 100)

## 3 调试模式

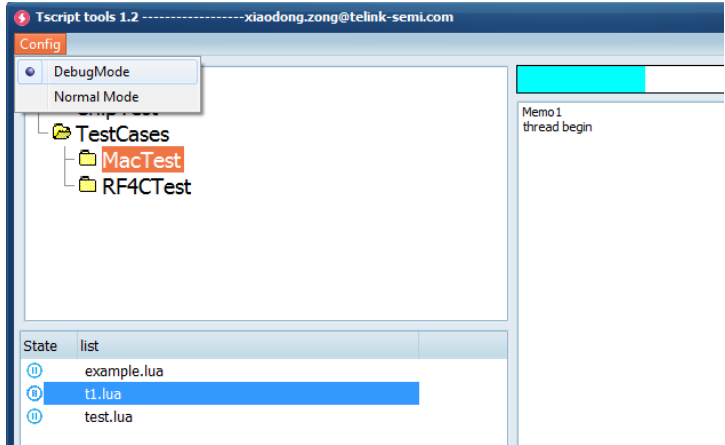
脚本运行有两种模式 debug 模式和 normal 模式

debug 模式运行速度较慢,但是可以加入断点,可以暂停脚本运行,可以强制停止脚本.

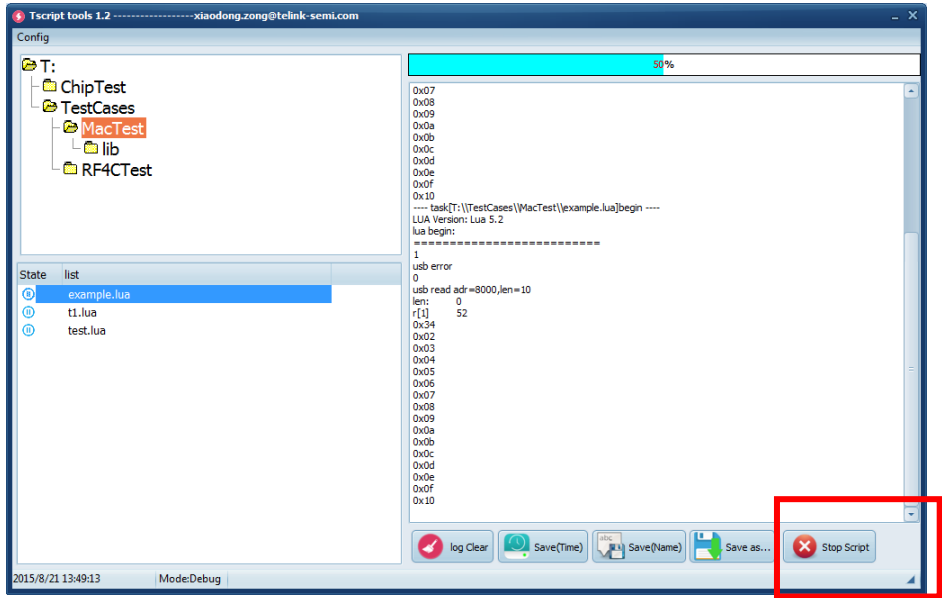
目前 Tscript 工具的 debug 模式只支持强制停止脚本功能.

normal 模式脚本可以全速运行.

建议在 debug 脚本时,用 debug 模式,这样可以在脚本进入死循环时,强制结束脚本.  
模式的选择如下所示:



在 debug 模式下,脚本运行时,可以通过下面的按键来强制脚本结束:



## 4 编辑脚本

可以使用任何一款文本文件编辑器写脚本,只要扩展名为\*.lua 就可以了.  
在界面中,可以通过鼠标右键点击脚本名字选择 edit,用文本编辑器打开脚本, 文本编辑器在 Tscript.ini 文件中设置.

