

# 基于 Tilcon 的综合导航信息处理装置界面设计

姜朝宇<sup>1</sup> 张巍<sup>1</sup> 周春红<sup>2</sup>

(1.海军驻葫芦岛四三一厂代表室, 辽宁 葫芦岛 125000;

2.合肥腾飞职业技术学校, 安徽 合肥 230000)

**摘要** Tilcon是一个支持多种操作系统平台下理想的图形用户界面开发工具。针对传统综合导航信息处理系统用户图形界面粗糙、操作复杂、源代码繁琐、维护困难等问题, 本文在对Tilcon的结构、特点、以及在VxWorks操作系统中的应用等方面进行深入分析的基础上, 采用Tilcon对原系统软件进行了改进设计。工程应用实践表明, Tilcon功能强大、使用方便、代码可移植性好、开发周期短, 是理想的嵌入式系统图形界面开发解决方案。

**关键词** 综合导航信息处理系统; 用户图形界面工具; VxWorks; Tilcon

## 1 引言

VxWorks 是高性能嵌入式实时操作系统。它以良好的可靠性和卓越的实时性被广泛地应用在通信、军事、航空、航天等高精尖技术及实时性要求极高的领域。但在图形界面显示方面相对薄弱, 较普遍的用法是采用风河公司提供的WindML 媒体库, 其图形功能使用 UGL 组件, 但是代码繁琐、效率较低、开发高质量图形界面相当困难<sup>[1]</sup>。针对嵌入式图形界面开发面临的困难, 出现了一些第三方的图形开发支持软件, 典型的有 Tilcon、Zinc、MiniGUI 等。由于 Zinc 的不稳定性, Tornado2.2 以上已经不再对其进行支持, 而 MiniGUI 的 API 封装的不够好, 不便于移植。由加拿大 Tilcon 软件公司设计的 Tilcon 图形开发工具集成大量成熟控件, 用拖动控件的方式构造自己的图形应用, 且与 Tornado 集成开发环境之间形成无缝连接, 非常适用于 VxWorks 系统下的图形界面开发<sup>[2, 3, 9]</sup>。

本文以某综合导航信息处理系统界面开发为背景, 重点探讨WindRiver VxWorks 5.5 下利用Tilcon 进行窗口图形界面的改进设计方法。

## 2 嵌入式图形开发工具 Tilcon

### 2.1 Tilcon 组成

Tilcon 的开发环境主要由三部分组成: Tilcon 嵌入式引擎 (EVE)、与平台无关的 API 接口 (API) 和可视化的界面设计环境。

嵌入式引擎是一个小巧的、以事件为驱动方式的内核。

它与用户的应用程序相独立, 负责处理用户所有的事件。API 是 Tilcon 提供的功能函数调用接口, 它为用户的应用程序与嵌入式引擎建立了一种通讯机制。它支持动态地创建对象, 提供对对象属性所有的控制, 同时保持使用上简单、方便。它本身也是使用 Tilcon 自己的功能函数开发的。可视化的界面设计环境为用户提供了丰富的控件, 用户只需通过拖动控件、设置控件的显示和运行属性的方式组建自己的图形界面, 用户的应用图形界面完成后, 存储为数据资源文件。在目标机环境中, 嵌入式引擎负责读取数据源文件 (即在可视化的界面设计环境中生成的\*.twd 文件), 绘制出相应的图形, 同时还可通过 TCP/IP 与其它目标机的嵌入式引擎相联系<sup>[4, 5]</sup>。其组成原理图如图 1 所示。

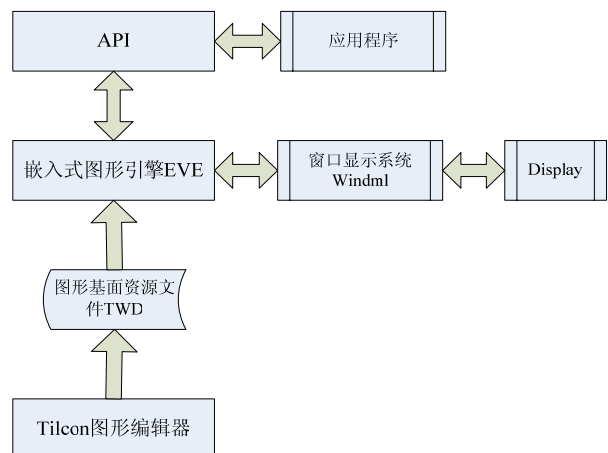


图 1 Tilcon 组成原理图

## 2.2 Tilcon 原理

Tilcon 工作原理如图 1 所示，用户使用 Tilcon 提供的可视化界面设计工具构建图形界面，图形界面要素以二进制形式保存到 TWD 文件中，这些资源文件是完全平台无关的，用户程序使用 Tilcon API 函数命令图形引擎装载（twd 文件）并控制图形引擎完成图形界面显示。同样，Tilcon 提供的 API 能兼容多个通用平台，支持应用程序的灵活移植。当一个 GUI 事件产生时（例如单击一个按钮），图形引擎接收到事件后，图形引擎根据事件类型决定直接处理或通知用户程序处理该事件。另一方面，当用户程序（控制进程）发送一系列指令（API 命令）给 Tilcon 图形引擎的命令队列，这些命令是异步的，用户程序不需要等待命令处理完毕（图形显示被更新）就可以直接返回到自己的主要任务中，图形引擎将负责完成这些命令的解释处理<sup>[6]</sup>。

## 3 基于 Tilcon 的嵌入式图形开发方法

### 3.1 Tilcon 的裁剪与配置

Tilcon 提供了图形引擎定制工具 Scalable，利用该工具用户可以灵活选择工程应用中必需的组件，除去不必要的组件并链接生成较小的 Tilcon 内核，减少 Tilcon 应用程序对系统空间的需求，这对嵌入式应用系统来说是非常有意义的<sup>[7]</sup>。具体步骤包括：在 Windows 操作系统上安装好 Tilcon 之后，在目录\tilcon\utilities\scalable 启动配置工具 Scalable，选择 VxWorks5.5 系统平台和 Pentium 处理机，并在控件选项中选择所需要的控件，工具会自动生成相应的 Makefile，点击创建按钮后工具将链接相关组件并生成相应的 Tilcon 的目标引擎 tlncore.o 和 tlnapi.o。

### 3.2 Windml 的配置与编译

WindML 是 Wind River 公司推出的多媒体支持库，是 Tilcon 在 VxWorks 系统上工作运行的基础。在使用 Tilcon 之前必须先配置编译 WindML。在配置 WindML 时，各选项可依据实际工程需要进行选择，但是为满足 Tilcon 执行条件，以下配置必须选择<sup>[8]</sup>：

- 在 Build 菜单栏中，“Build WindML object”和“Build WindML examples”选项要选上。
- 在 Graphics 菜单栏中，界面设置中的分辨率应该定为“800\*600”。
- 在 Bitmap 菜单栏中，“Include Unicode Support”选项要选上，并且要包含所有字体支持。

### 3.3 VxWorks 配置和编译

配置和编译好 WindML 和 Tilcon 后，根据工程应用需要在 VxWorks 工程中对 VxWorks 进行配置，需将 C++

Components、POSIX Clocks、POSIX Message Queues、POSIX Semaphores、POSIX Timers 选择包含进 VxWorks 工程中，编译生成 VxWorks 映像。另外，用户可以使用 Tornado 工程管理工具把 WindML 和 Tilcon 连接到 VxWorks 映像里。对于 Tilcon 图形界面的开发，至少需要将 WindML 的相关组件 2D graphics 和 complete 2D library 选择包含进 VxWorks 工程中，并将 Tilcon 内核目标文件 tlncore.o 和 tlnapi.o 连接到工程中。

### 3.4 图形界面设计

Tilcon 窗口的开发主要通过 Tilcon 图形编辑器来实现，Tilcon 图形编辑器是一组集成的设计工具用来辅助用户完成图形界面的设计。在使用 Tilcon 设计工具（Tilcon 图形编辑器）构建图形界面的屏幕显示时，使用鼠标以“拖-放”的方式从工具栏添加用户界面对象，并点击鼠标右键打开属性框编辑对象的属性，所有的对象都可以通过填写对象属性框的方式进行编辑，所有属性的改变都会立刻在开发环境中反映出来，通过属性项的设定调整对象的所有外观效果（例如颜色、标签、边框、材质和字体等）。通过对象的属性项还可以指定对象的行为，包括大多数功能、事件响应方式，初始状态，文本换行方式，显示焦点，帮助对话框，菜单，鼠标指针和快捷键等。编辑完成之后，设计的图形界面保存为 twd 文件。如果需要用户可以点击开发环境的测试按钮来检验整个图形界面的运行效果，另外，Tilcon 可以使用用户指定的模拟数据在测试状态下驱动各个对象。

### 3.5 应用程序的软件设计

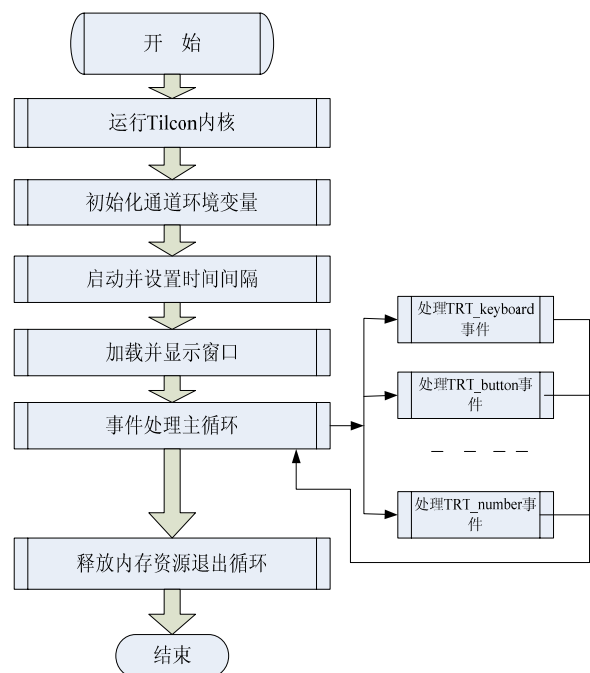


图 2 软件设计流程图

Tornado 集成开发环境下基本编程思路如图 2 所示，其概述如下：首先调用 TRT\_Start 应用程序接口函数运行 Tilcon 内核，此时 VxWorks 与 Tilcon 之间的信息通道打开，然后开始图形用户界面窗口初始化操作并进行加载显示，随后进入主框架 While (1) 循环区域同时 TRT\_GetInput 函数处于等待状态并实时接收来自于 Tilcon 矢量引擎所传递的界面操作事

件，将其通过 Switchcase 条件语句结构分发给相应的处理子函数，如果接收到终止信息则清除图形用户界面用于释放内存，最后调用 TRT\_Exit 函数完成退出操作。

依据综合导航信息处理系统需求分析，其人机交互界面应实现所有人机交互功能，因此对于其图形界面设计我们采用模块式设计，其功能操作示意图如图 3 所示。

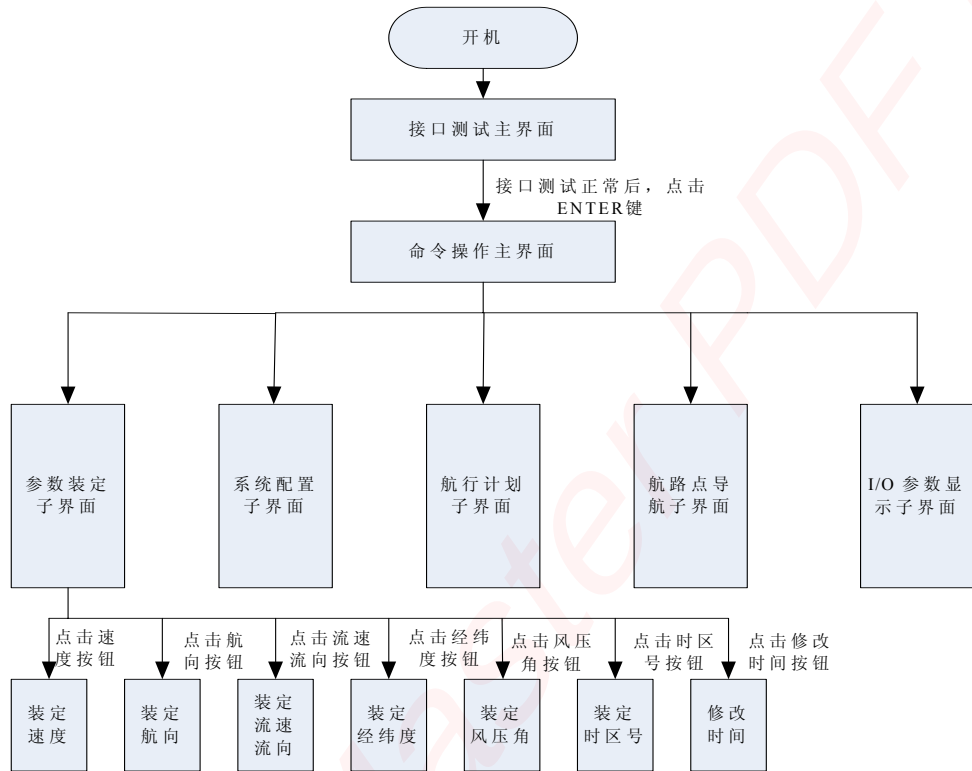


图3 功能操作示意图

下面是本工程主要程序：

```
#include <time.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <tilcon/TRTAPI.h>
#ifdef CC_TRT_VXWORKS
#define TRT_OS_ENV TRT_VXWORKS
#endif
pid_t TRT_cid; TRT_Receive Data rec_data;
char *main_window_file = MAIN_WINDOW_FILE;
char main_window_id[TRT_MAX_ID_LENGTH];
// 主程序
int main (int argc, char argv[ ])
{
    long errorcode = 0 ;
```

```
int c;
long running ;
int ContinueLooping = TRUE;
TRT_StartData StartData ;
// 设定系统环境
#ifdef CC_TRT_VXWORKS
StartData.OsEnv = TRT_VXWORKS ;
// The operating system this app is running under.
#endif
StartData.Display = NULL ;
StartData.IPAddr = NULL ;
StartData.Userprog = APPNAME;
StartData.AppName = USERPROG;
StartData.Flags = NULL ; // For now set to NULL
running = TRT_Start (&StartData);
// 启动 Tilcon 图形引擎
```

```

errorcode = TRT_Start ( &StartData );
TRT_cid = StartData. TRT CID ;
if ( errorcode )
{
    printf ( "Cannot Start Runtime\n" );
    exit ( 0 );
}
// 加强所有 API 命令同步
errorcode = TRT_Debug ( TRT_cid, 3 );
if ( errorcode ) ExitBox ( "Cannot run TRT_Debug" );
// Setup timer hint s for 1/ 10th second
errorcode = TRT_TimerHint_Enable ( TRT_cid, 2 ); //
2 × 50 ms//
if ( errorcode ) ExitBox ( "Cannot start timerhint s" );
// 装载 Main Window
errorcode = TRT _WindowLoad ( TRT_cid , main
windowfile );
if ( errorcode ) ExitBox ( " Cannot load the Main Window !
File NOT found" );
errorcode = TRT_GetWindowID ( TRT_cid, main window
id );
if ( errorcode ) ExitBox ( "Cannot Get Main Window ID" );
// 显示 Main Window
errorcode = TRT_WindowDisplay ( TRT_cid, main window
id );
if ( errorcode ) ExitBox ( "Cannot Display Main Window" );
// 主循环
while ( ContinueLooping )
{
    // Wait for a notification
    c = TRT_Get_Input ( NULL, 0, NULL, 0, &rec data,
TRT BLOCK );
    switch ( c )
    {
        // Received a standard notification
        case 0 :
            {
                // what type of object is notifying us
                switch ( rec_data. code )
                {
                    Case: TRT_window :

```

```

if ( rec data. state == TRT_window_quit )
    {
        ContinueLooping = FALSE;
    }
break ;
case TRT_button :
    break ;
default : break ;
}
}
break ;
case 1 :
    break ;
case - 1 :
    break ;
default : break ;
}
}
TRT_WindowDelete ( TRT_cid, main window id ); //Close
Window
TRT_Exit ( TRT_cid ); // Notify Runtime to exit return ( 0 );
其部分实例效果图如图 4 所示：

```



图 4 部分实例效果图

### 3.6 窗体的切换与刷新

图形的动态刷新是图形技术的要点和难点问题。以本开发工程为例，画面上航迹块的移动、数据表格的更新、指示仪表指针的摆动等都需要根据具体的数据进行实时刷新和显示。通常的动态刷新方法是把刷新对象原先的图形用背景覆

( 下转第 12 页 )

```
#define rACE0_DLM (*(volatile unsigned char *)  
(ACE0_BASE+0x01)) // ACE0 波特率发生分频因子高八  
位, LCR.7 (DLAB) =1, 可读写
```

## 5 结束语

基于 TL16C554 的嵌入式系统的多串口扩展, 其硬件实现电路简单, 可靠性高, 使用经济性好。在笔者参与的智能电表多路采集控制器中使用此方案, 运行稳定, 证明它是一种可行的嵌入式系统串口扩展方案。

## 参考文献

[1] TL16C554 Data manual. Texas Instrument

Incorporated

[2] S3C44B0X USER MANUAL.SAMSUNG  
ELECTRONICS

[3] ADM2483 Data manual. Analog Device Inc

[4] 刘彦文. 基于 ARM7TDMI 的 S3C44BOX 嵌入式微  
处理器技术

收稿日期: 4 月 6 日 修改日期: 4 月 19 日

作者简介: 戚建宇 (1978-) 男, 硕士研究生, 研究方  
向: 信息处理; 何松 (1980-), 女, 硕士研究生, 研究方  
向: 信息处理。

(上接第 8 页)

盖掉再重新画, 这种方法会耗费大量的系统资源, 特别对于持续更新的对象, 显示效果上会有闪屏或托尾现象, 非常不可取。现在这个问题可以方便的利用 Tilcon 内嵌的双缓冲区技术, 屏幕下重绘方式来解决。要刷新的对象或区域首先被生成不在屏幕上显示的图像, 然后此图像再被复制到屏幕上。这种方式可以在设计界面时在图形编辑器的 Redraw Option 中选择 Offscreen 固定下来, 也可在程序运行时用 API 函数命令设置。本项目中需要实时刷新的对象直接在窗体上设为 Object Offscreen, 背景区域设为 Area Offscreen, 这样得到了比较清晰平滑的刷新效果。

## 4 结论

Tilcon 以它独特的二进制图形引擎的方式解决了嵌入式实时系统 VxWorks 中图形界面开发中的难题。它为用户提供的可视化界面设计环境简化了用户的编程, 提高了用户项目开发的效率。Tilcon 在功能、性能方面都能满足基于 VxWorks 的实时系统窗口图形界面开发要求, 是理想的嵌入式系统图形界面开发解决方案。

## 参考文献

[1] WindRiver Systems Inc. VxWorks Programmer' s Guide

5.5[M]. USA, 2002

[2] 刘炳锋, 李江红. 基于 Tilcon 的某武器显控系统界面  
设计[J]. 计算机测量与控制, 2008, 16 (8): 1153-1154

[3] 王文. 基于 PC\_104 的嵌入式操作系统研究[A]. 哈尔滨  
工程大学硕士学位论文[C], 哈尔滨工程大学, 2009

[4] 李阳, 黄浩华, 刘晓亮. 嵌入式图形系统 Tilcon 及应  
用研究[J]. 计算机与数字工程, 2008 (2): 110-112

[5] Tilcon Software Ltd. Tilcon Interface Builder Tutorial.  
2005

[6] Tilcon Software Ltd. Tilcon Programmer' s Guide [Z].  
CA, 2005

[7] Tilcon Technology & Architecture Guide Manual  
[EB/OL]. <http://www.tilcon.com>

[8] WindRiver Systems Inc. WindML SDK Programmer' s  
Guide 3.0[Z]. USA, 2002

[9] 张加林, 李松, 汪高武等. 基于 Tilcon 的 VxWorks 图  
形界面开发技术[J]. 指挥控制与仿真, 2007, 29 (5): 1002103

作者简介: 姜朝宇 (1979-), 男, 控制理论与控制工  
程专业, 研究方向为导航、制导与控制。