

基于 KDevelop 的 MiniGUI IDE 的设计与实现

Design and Implementation the KDevelop Based IDE Solution for MiniGUI Application Development

陈 雷 王文杰 (中国科学院研究生院 北京 100049)

摘 要: MiniGUI 是一款开放源码的、针对嵌入式设备的图形界面支持系统,越来越多的被作为产品开发的 GUI 基础。然而,在 Linux 环境下,MiniGUI 缺乏一个可视化的集成开发调试环境,给开发人员带来很大的不便。针对这一问题,该文分析了 KDevelop 开发环境的特点,探讨了 Linux 环境下利用 KDevelop 环境进行 MiniGUI 程序的开发和调试的可行性,提出了针对 KDevelop 的改造方案,并给出了详细的设计与实现过程。

关键词: Linux MiniGUI KDevelop QT Designer 集成开发环境

1 引言

随着嵌入式 Linux 操作系统应用的成熟和普及,越来越多的项目和产品开始用其作为底层的基本操作系统。而作为基本的用户交互界面,图形应用程序开发的需求也日益增加。

MiniGUI 是一款中国人开发的,轻量级图形界面支持系统。它具有体积小,运行速度快,对中文的天然的良好支持,对系统资源要求相对较低,可配制性较高等特点。广泛应用在工业仪表、医疗仪器、通讯终端、电视多媒体、军工等行业。

2 需求的提出

然而,传统的 MiniGUI 程序开发和调试模式是基于文本编辑器以及命令行调试工具的。缺乏一个集成的可视化的开发和调试环境是困扰 MiniGUI 进一步普及和推广的问题之一。

Linux 环境下 MiniGUI 程序的典型开发模式如下:

- (1) 使用文本编辑器编写应用程序的源文件。
- (2) 编写 Makefile 文件用以组织和配置项目以及编译链接等信息。
- (3) 使用 GNU automake, autoconf, GCC 等工具编译源文件,生成可执行文件。
- (4) 在字符终端中使用 gdb 运行/调试编译好的图形应用程序,这会在终端上直接显示出程序的图形界面。
- (5) 如果这时需要切换到 gdb 的界面进行相应的调试,则需要按特殊的按键组合在 gdb 图形界面之间进行切换。

上述开发过程对于刚刚接触类 Linux 操作系统的开发人员来说有以下弊端:

- (1) 编辑程序时,不容易适应字符终端的使用
- (2) 自己需要组织项目的结构,设置编译环境变量,选项以及编写 Makefile 等等。
- (3) 程序源代码编写的过程中,无法预先看到窗口、控件的大致效果,只有在运行时才可以看到。
- (4) 基于字符界面的调试器(gdb)没有图形化的直观简便。

基于以上考虑,为 MiniGUI 提供一个基于图形界面的集成开发环境(IDE)成为需要解决的问题。

3 选择 KDevelop 的可行性

Kdevelop 是一款广受欢迎的基 KD 环境的可视化集成开发调试环境。其主要特点包括:支持各种项目模板(例如,QT、GTK 等);支持多种语言的代码编辑器(C/C++ 等);可以与调试器配合实现基于图形界面的源代码级调试。其与 QT Designer 配合更可以实现所见即所得的程序开发。我们考虑在其基础上来开发 MiniGUI 的 IDE。考察改造的可行性需要考虑以下几个方面:

* 工程项目的组织

KDevelop 遵循并通过与 GNU 的 autoconf、auto-make 配合来实现源代码的组织。作为开源软件项目,MiniGU 程序的项目组织也天然地使 autoconf 和 auto-make 规范组织其项目。

* 编程语言支持

目前,MiniGUI 所使用的语言为标 C。KDevelop 对于 C/C++ 的支持很好。

* 图形窗体开发

KDevelop 与 QT Designer 相配合可以实现 QT 程序的所见即所得开发。QT Designer 是一款用于开发 QT 图形程序的界面创建工具,其操作风格类似于 Microsoft Visual Studio 的对话框界面生成器,同时增加了 QT 自有的特性。其工作原理如图 1 所示:

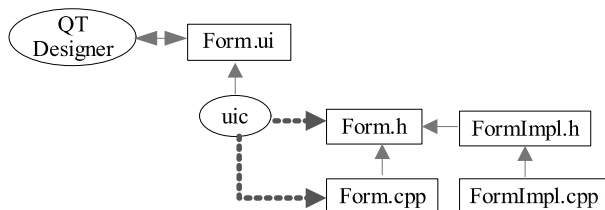


图 1 QTDesigner 的工作原理

KDevelop 与 QTDesigner 协作的典型过程如下:

1) KDevelop 调用 QTDesigner 完成对话框窗口的定制,生成中间文件(.ui 文件),其存储格式为 XML。

2) KDevelop 使用 uic 程序对中间文件进行解析、产生相应的 .cpp 和 .h 文件,并添加到当前项目中。

在此,我们可以改造 uic 程序,使之能够将针对 QT 库特性的 .ui 文件转化为 MiniGUI 相应的各种控件,并且生成相应的 .rc(控件相关信息)、.h 和 .c(用户逻辑部分)文件。如图 2 所示:

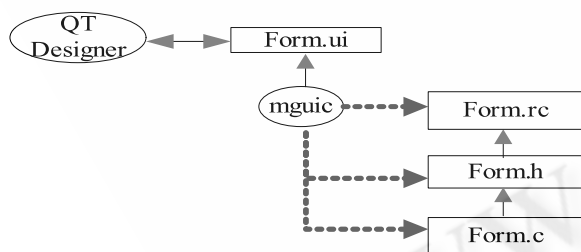


图 2 修改之后 uic 与 QTDesigner 的配合

* 程序编译

KDevelop 支持编译 MiniGUI 程序所需要的编译以及连接工具(GNU gcc, ld 等等)。

* 图形程序运行和调试环境

MiniGUI 程序需要运行在 FrameBuffer 之上。KDE 环境提供了 QVFB 用来在图形环境中提供模拟的 FrameBuffer 设备。目前,MiniGUI 已经支持 QVFB 作为图形输出设备。

4 实现方案

(1) 准备 MiniGUI 默认开发环境

* 从 MiniGUI 网站 (<http://www.minigui.com/index.php?id=minigui-open-source-version&l=1>) 下载并安装 MiniGUI 开发包 (MiniGUI-dev)。

* 修改 MiniGUI 配置文件,使其图形设备指向 QVFB。

* 可选:安装 MiniGUI demonstration 演示包 (MDE 或者 samples)。

(2) 修改 KDevelop 源代码,并生成可执行文件,用其替换原有的 KDevelop 程序。

* 添加创建 MiniGUI 工程类别部分的逻辑以及相应项目向导。

* 添加与 MiniGUI 相关菜单

(3) 制作用以创建 MiniGUI 的 KDevelop 项目支持文件

其中包括 HelloWorld 框架程序,autoconf、automake 相关配置文件以及 MiniGUI 相关的配置文件,如下显示了 MiniGUI 用于 KDevelop 的应用程序模板的目录组织结构:

```

mgtemplate/
  Makefile.am
  Makefile.dist
  Configure.in
  admin/
    //KDevelop admin script files
    ...
  src/
    Makefile.am
    hello.c
    hello.h
    
```

其中,

* Makefile.am、makefile.dist 为 GNUautoconf、automake 相关的项目组织文件。

* Configure.in 中包含了 MiniGUI 相关的编译以及和连接选项。

* admin 目录中是 KDevelop 用于项目组织和管理的各种脚本文件(需改造)。

* src 目录是 MiniGUI 的 HelloWorld 主框架程序。

* hello.c 是一个样例程序,运行时会出现一个 MiniGUI 主窗口,并在其中显示“Hello World!”。开发

者可根据需要修改这个程序,并且在该项目框架的基础上发展自己的项目。

(4) 修改 uic 程序编译生成支持 MiniGUI 对话框转换的 mguic 程序。改动主要包括三个部分:

- * 用户接口部分
- * XML 语法分析部分,用来解析 QT Designer 所生成的 DOM (Document of Module) 树

MiniGUI 程序生成部分,其中由于开发者可能会反复修改对话框(位置、大小、控件及属性等),导致系统反复生成 .h 和 .c 程序,开发者所编写的逻辑会被覆盖掉。为解决以上问题,我们生成一个资源文件 .rc 用来保存以上信息,而将开发者定制的逻辑部分放在 .c 文件中。此外,还需要过滤掉其中 QT 的特有语法特征,将 QT 图形元素(各控件的属性)信息(例如,位置信息,Name, Caption,等等)转换成相对应的 MiniGUI 图形元素信息。图 3 表示了 mguic 的流程,表 1 所示,是 MiniGUI 与 QT 常用控件的对应表:

表 1 QT 与 MiniGUI 控件对应表

minigui 控件	QT 控件
CTRL_STATIC	QLabel
CTRL_BUTTON	QPushButton
CTRL_BUTTON	QRadioButton
CTRL_BUTTON	QCheckBox
CTRL_LISTBOX	QListBox
CTRL_PROGRESSBAR	QProgressBar
CTRL_EDIT	QLineEdit
CTRL_MEDIT	QTextEdit
CTRL_STATIC	QGroupBox
CTRL_COMOBOX	QComboBox
CTRL_SPINBOX	QSpinBox

(5) 安装 QVFB 程序,从互联网上或者 MiniGUI 网站下载并编译安装 QVFB 程序。

5 基于新 KDevelop IDE 的 MiniGUI 程序开发

(1) 新建 MiniGUI 项目

在新 KDevelop 环境下,选择菜单 -> 新建项目,选择 MiniGUI 项目其余选择默认即可。完成后,会生成 MiniGUI 的默认项目以及主程序,如图 4 所示:

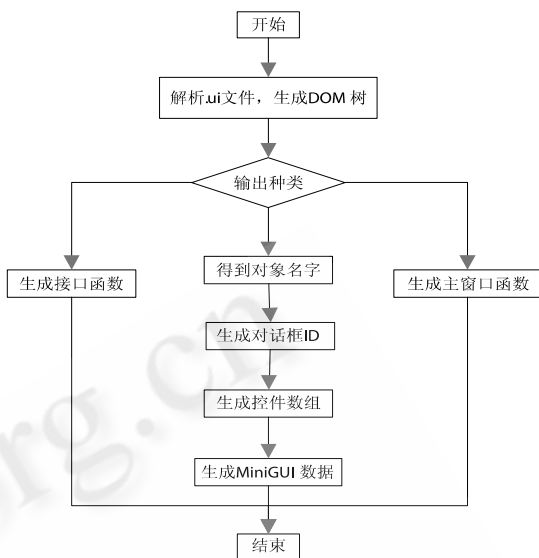


图 3 mguic 流程图

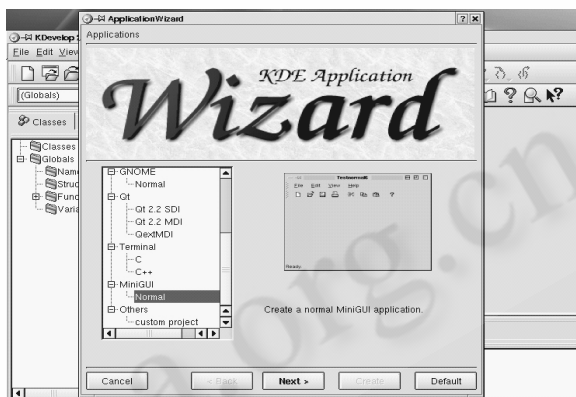


图 4 新建 MiniGUI 项目

(2) 编译

选择菜单,编译,如图 5 所示:

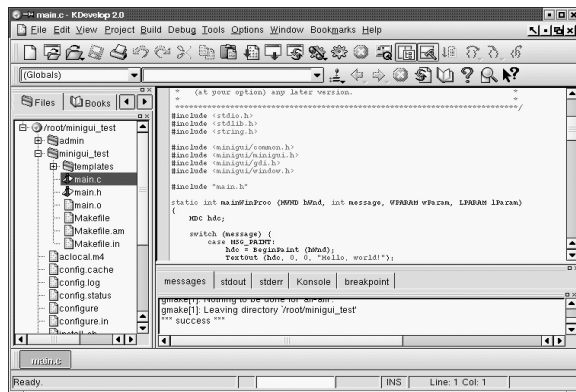


图 5 生成的 MiniGUI 项目框架的编译

(3) 运行

* 修改 MiniGUI 配置文件使其图形输出设备和输入设备指向 QVFB。其中关于 QVFB 的配置可根据自己需要而定。

```
[system]
#GAL engine
gal_engine = qvfb
#IAL engine
ial_engine = qvfb
mdev = /dev/mouse
mtype = IMPS2
( 如为 FC3,则改为 mdev = /dev/input
    mtype = mouse )
....
[qvfb]
defaultmode = 800x600 - 16bbp
display = 0
....
```

* 启动 QVFB, 将其分辨率设置为上一步中 MiniGUI 配置文件里面设定的值。

* 在 KDevelop 中, 菜单选择, 运行。如图 6 所示:

(4) 添加 MiniGUI 对话框

选择菜单, 新建对话框, 输入对话框名称。例如, MyDialog。此时 QT Designer 会被 KDevelop 启动, 开始设计、编辑对话框界面。如图 7 所示:

当关闭 QTDesigner 之后, KDevelop 使用 mguic 处理 .ui 文件, 并且生成 .h 以及 .c 文件。用户可以在项目文件视图中看到自动生成的 MyDialog.c 和 MyDialog.h。用户可在其基础上自行添加应用逻辑, 在主窗口代码中 #include MyDialog.h, 进而在适当的地方 (例如主程序中) 调 MyDialog 对话框

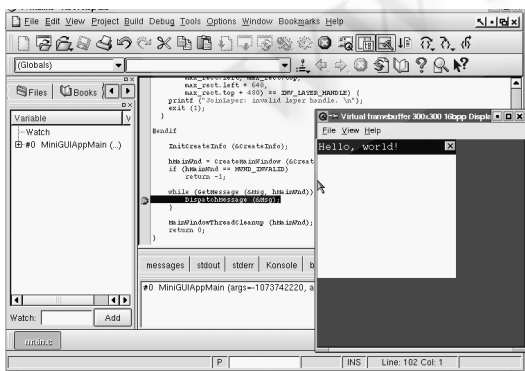


图 6 MiniGUI 程序调试

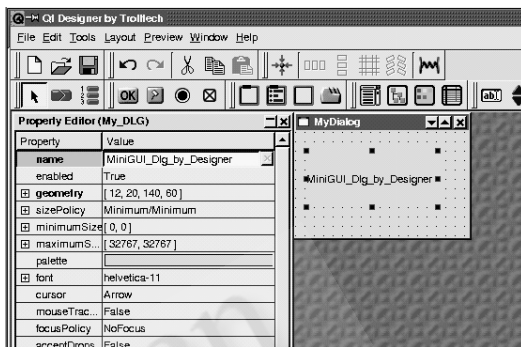


图 7 QT Designer 设计 MiniGUI 对话框

(5) 调试

在 KDevelop 中, 浏览 MiniGUI 程序源代码, 并设置断点。菜单选择, 运行, 程序执行到所设置断点处会自动暂停, 并且调试光标会停留在该代码行处。此时可以进行单步执行, 变量、数据查看与监视等调试动作, 如图 8 所示:

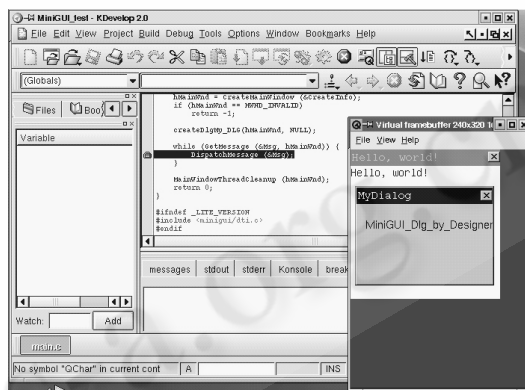


图 8 调试所生成的对话框代码

参考文献

- 1 魏永明. MiniGUI 源代码包及文档. <http://www.minigui.com/index.php?id=minigui-doc-dev-package&L=1>, 2003.
- 2 孙家广, 杨长贵. 计算机图形学. 北京: 清华大学出版社, 1999: 70 - 120.
- 3 KWall, M.Watson, M.Whitis, et al. GNU /Linux 编程指南. 王勇, 等. 北京: 清华大学出版社, 2000, 30 - 80.
- 4 Nolden Ralf. The KDevelop Programming Handbook. , <http://kdevelop.org/1.3/doc/programming/index.html>, 1999.