MiniGui 开发包指引

Version 1.00

REVISION HISTORY

Revision No.	Description	Date
1.00	• Initial release	11/30/2018

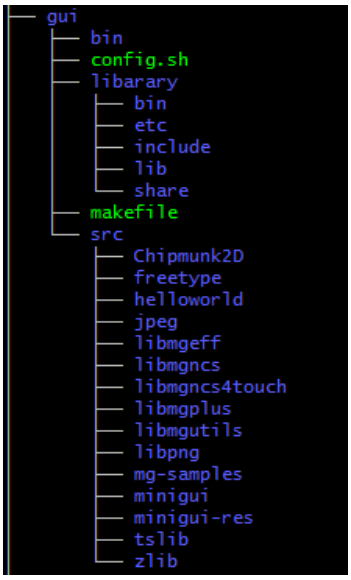
TABLE OF CONTENTS

1. 软件包概况

1.1. MiniGui 资源

参考官网 http://www.minigui.com/

1.2. 软件包概略图



1.3. 软件包清单

基本库 :freetype tslib zlib jpeg libpng Chipmunk2D minigui libmgplus libmgutils libmgeff libmgncs
图片资源:minigui-res
演示 APP:libmgncs4touch mg-samples helloworld
软件来源：http://www.minigui.com/zhcn/download (MiniGui 3.2.0)
https://github.com/slembcke/Chipmunk2D
(commit: 6b5b827a1c739437ba191dde5cf2446648417b82)
https://github.com/kergoth/tslib
(commit: faf2afd088b3dc7c09c532bc2aeb915cfd05f09c)

2. 编译

2.1. 工具链

gcc 工具链版本与参数

```
Target: arm-linux-gnueabihf
Configured with: /cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabihf-linux/.build/src/gcc-linaro-4.8-2014.04/configure
--build=i686-build_pc-linux-gnu --host=i686-build_pc-linux-gnu --target=arm-linux-gnueabihf
--prefix=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabihf-linux/install
--with-sysroot=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabihf-linux/install/arm-linux-gnueabihf/libc
--enable-languages=c,c++,fortran --enable-multilib --with-arch=armv7-a --with-tune=cortex-a9 --with-fpu=vfpv3-d16 --with-float=hard
--with-pkgversion='crosstool-NG linaro-1.13.1-4.8-2014.04 - Linaro GCC 4.8-2014.04' --with-bugurl=https://bugs.launchpad.net/gcc-linaro
--enable-__cxa_atexit --enable-libmudflap --enable-libgomp --enable-libssp
--with-gmp=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabihf-linux/.build/arm-linux-gnueabihf/build/static
--with-mpfr=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabihf-linux/.build/arm-linux-gnueabihf/build/static
--with-mpc=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabihf-linux/.build/arm-linux-gnueabihf/build/static
--with-isl=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabihf-linux/.build/arm-linux-gnueabihf/build/static
--with-cloog=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabihf-linux/.build/arm-linux-gnueabihf/build/static
--with-libelf=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabihf-linux/.build/arm-linux-gnueabihf/build/static
--enable-threads=posix --disable-libstdcxx-pch --enable-linker-build-id --enable-gold
--with-local-prefix=/cbuild/slaves/oorts/crosstool-ng/builds/arm-linux-gnueabihf-linux/install/arm-linux-gnueabihf/libc --enable-c99
--enable-long-long --with-mode=thumb --with-float=hard
Thread model: posix
gcc version 4.8.3 20140401 (prerelease) (crosstool-NG linaro-1.13.1-4.8-2014.04 - Linaro GCC 4.8-2014.04)
```

2.2. 编译软件包

2.2.1 编译命令

make clean： 清除全部的编译结果

make: 全部编译并安装

make libs: 编译所有的库

make apps:编译所有的 APP

以上所有的命令都会去执行一遍 config，但是 config 通常只需要执行一次，全部执行比较耗时，可以使用具体的编译命令编译需要的目标。

全部的编译选项：

```
all                 clean               jpeg_build          libmgeff_install    libmgplus_config    mg-samples_build    minigui-res_install
app_build           config              jpeg_clean          libmgncs4touch_build libmgplus_install  mg-samples_clean    tslib_build
app_clean           freetype_build      jpeg_config         libmgncs4touch_clean libmgutils_build   mg-samples_config   tslib_clean
app_config          freetype_clean      jpeg_install        libmgncs4touch_config libmgutils_clean  mg-samples_install  tslib_config
app_install         freetype_config     lib_build           libmgncs4touch_install libmgutils_config minigui_build      tslib_install
apps                freetype_install    lib_clean           libmgncs_build      libmgutils_install  minigui_clean       zlib_build
build               helloworld_build    lib_config          libmgncs_clean      libpng_build        minigui_config      zlib_clean
Chipmunk2D_build    helloworld_clean    lib_install         libmgncs_config     libpng_clean        minigui_install     zlib_config
Chipmunk2D_clean    helloworld_config   libmgeff_build      libmgncs_install    libpng_config       minigui-res_build   zlib_install
Chipmunk2D_config   helloworld_install  libmgeff_clean      libmgplus_build     libpng_install      minigui-res_clean
Chipmunk2D_install  install             libmgeff_config     libmgplus_clean     libs                minigui-res_config
```

2.2.2 编译结果

在默认的 makefile 配置下，编译安装完成之后，关键目标的位置如下：
配置文件: gui\libarary\etc
头文件 ：gui\libarary\include
库文件 ：gui\libarary\lib
可执行文件:

APP	路径	demo	资源	描述
libmgncs4touch	gui/src/libmgncs4touch/samples/.libs	btnnavbar	gui/src/libmgncs4touch/samples/res	
		commdraw		
		datepicker		
		exlist		
		hscrollview		
		iconflow		
		imgnavbar		
		itembar		
		name2value		
		newtrackbar		
		picker		
		settingtableview		
		switchbutton		
		tableview		
		timepicker		
mg-samples	gui/src/mg-samples/ctrl_dlg/	ctrl_dlg	gui/src/mg-samples/ctrl_dlg/res	
	gui/src/mg-samples/dbuff/	dbuff	gui/src/mg-samples/dbuff/res	
	gui/src/mg-samples/font_text/	font_text	gui/src/mg-samples/font_text/res	
	gui/src/mg-samples/graphics/	graphics	gui/src/mg-samples/graphics/res	
	gui/src/mg-samples/housekeeper/	housekeeper	gui/src/mg-samples/housekeeper/res	
	gui/src/mg-samples/mginit/	mginit	gui/src/mg-samples/mginit/res	
	gui/src/mg-samples/minesweeper/	minesweeper	gui/src/mg-samples/minesweeper/res	
	gui/src/mg-samples/notebook/	notebook	gui/src/mg-samples/notebook/res	
	gui/src/mg-samples/same/	same	gui/src/mg-samples/same/res	
helloworld	gui/src/helloworld	mginit		
		helloworld		

资源文件: gui/libarary/share/minigui

2.2.3 Sigmastar IPC 软件平台配置

2.2.3.1. Kernel

使用配置 `infinity5_ssc007a_s01a_spinand_defconfig`
使能 device driver 下的配置项目:

Enable TTY
Virtual terminal

```
        Performance monitor support  ----
    [ ] Reliability, Availability and Serviceability (RAS) features  ----
    [*] Enable TTY
    [*]   Virtual terminal
    [ ]     Enable character translations in console
    [ ]     Support for console on virtual terminal
    [ ]     Support for binding and unbinding console drivers
```

使能 device driver > HID support 下的配置项

HID bus support

Generic HID drivers

```
        {*} HID bus support
        [ ]   Battery level reporting for HID devices
        [ ]   /dev/hidraw raw HID device support
        < >   User-space I/O driver support for HID subsystem
        <*>   Generic HID driver
              Special HID drivers  --->
            USB HID support  --->
            I2C HID support  --->
```

使能 device driver > HID support 下的配置项

USB HID transport layer

```
            <M> USB HID transport layer
            [ ] PID device support
            [ ] /dev/hiddev raw HID device support
                USB HID Boot Protocol drivers  --->
```

使能 device driver > generic driver options 下的配置项

Userspace firmware loading support

```
        [*] Maintain a devtmpfs filesystem to mount at /dev
        [*]   Automount devtmpfs at /dev, after the kernel mounted the rootfs
        [ ] Select only drivers that don't need compile-time external firmware
        [ ] Prevent firmware from being built
        -*- Userspace firmware loading support
        [*]   Include in-kernel firmware blobs in kernel binary
        ()    External firmware blobs to build into the kernel binary
        [ ] Fallback user-helper invocation for firmware loading
        [ ] Allow device coredump
```

使能 device driver > input device support >Touchscreens 下的配置项

Goodix I2C touchscreen

```
            < >   EETI eGalax multi-touch panel support
            < >   EETI eGalax serial touchscreen
            < >   Fujitsu serial touchscreen
            <*>   Goodix I2C touchscreen
            < >   Ilitek ILI210X based touchscreen
            < >   Gunze AHL-51S touchscreen
            < >   Elan eKTF2127 I2C touchscreen
```

2.2.3.2. Project

在对应的 misc_mod_list_late 文件末尾添加 mstar_fbdev.ko

3. DEMO 运行

3.1. 配置运行环境

3.1.1 运行库

将需要的动态库 copy 至板端合适的位置,例如/customer 分区,配置 LD_LILBARARY_PATH

3.1.2 资源文件

将需要运行的 APP 的资源文件 copy 到板端合适的位置，例如/customer 分区。

3.1.3 临时文件

MinGui 运行需要一个/var/tmp 的临时文件夹，需要事先创建

3.1.4 配置文件

将编译生的配置 copy 到/etc/ 文件夹下面

3.1.5 配置文件参数介绍

3.1.5.1. MiniGUI.cfg

一个典型的触屏方案 system 项配置参数

```
[system]
# GAL engine and default options
gal_engine=fbcon
defaultmode=800x480-16bpp

# IAL engine
ial_engine=tslib
mdev=/dev/input/event0
mtype=none
```

配置 MiniGui 使用的图形引擎，fbcon 为使用 linux 的 framebuffer 设备

```
gal_engine=fbcon
```

配置 MiniGui 的画布大小以及颜色深度

```
defaultmode=800x480-16bpp
```

配置 MiniGui 的输入引擎，tslib 是一个第三方的读取 touchscreen 输入设备节点的库。
引擎配置成 tslib 将使用这个库来读取触屏输入。

```
ial_engine=tslib
```

touchscreen 的输入设备节点

```
mdev=/dev/input/event0
```

fbcon 默认需要配置的项目

```
[fbcon]
defaultmode=800x480-16bpp
```

配置资源的路径

```
[cursorinfo]
# Edit following line to specify cursor files path
cursorpath=/customer/minigui/res/cursor/
```

```
[resinfo]
respath=/customer/minigui/res/
```

3.1.5.2. ts.conf

tslib 可以根据配置对 touchscreen 的输入进行过滤

在目前的配置下必须开启的项目为：

```
module_raw input
```

如果开启额外的项目，需要配置 tslib 的编译选项打开对应的功能

```
# Access plugins
#################

# Uncomment if you wish to use the linux input layer event interface
module_raw input

# For other driver modules, see the ts.conf man page


# Filter plugins
#################

# Uncomment if first or last samples are unreliable
# module skip nhead=1 ntail=1

# Uncomment if needed for devices that measure pressure
module pthres pmin=1

# Uncomment if needed
# module debounce drop_threshold=40

# Uncomment if needed to filter spikes
# module median depth=5

# Uncomment to enable smoothing of fraction N/D
# module iir N=6 D=10

# Uncomment if needed
# module lowpass factor=0.1 threshold=1

# Uncomment if needed to filter noise samples
```

```
module dejitter delta=100

# Uncomment and adjust if you need to invert an axis or both
# module invert x0=800 y0=480

# Uncomment to use ts_calibrate's settings
module linear
```

3.1.5.3. fbdev.ini

参考 framebuffer 配置文档对/config/fbdev.ini 做出配置


Framebuffer配置.pdf

一份适用于当前软件包的配置

```
[FB_DEVICE]
FB_HWLAYER_ID = 1
FB_HWWIN_ID = 0
FB_HWLAYER_DST = 3
FB_HWWIN_FORMAT = 5
FB_HWLAYER_OUTPUTCOLOR = 1
FB_WIDTH = 800
FB_HEIGHT = 480
FB_TIMMING_WIDTH = 800
FB_TIMMING_HEIGHT = 480

[LAYER_ZORDER]
LAYER_ZORDER0 = 0
LAYER_ZORDER1 = 1
LAYER_ZORDER2 = 2
LAYER_ZORDER3 = 3
LAYER_ZORDER4 = 4
```

3.2. 运行 demo

3.2.1 串流

运行 gui/bin/prog_rtsp 串出 sensor 到 panel 的视频流。
panel 以及 sensor 驱动工作需要参考其他文档，包括修改 prog_rtsp 的源码。
目前的 prog_rtsp 只支持 AT070TN94_800x480 这个 panel

3.2.2 mginit

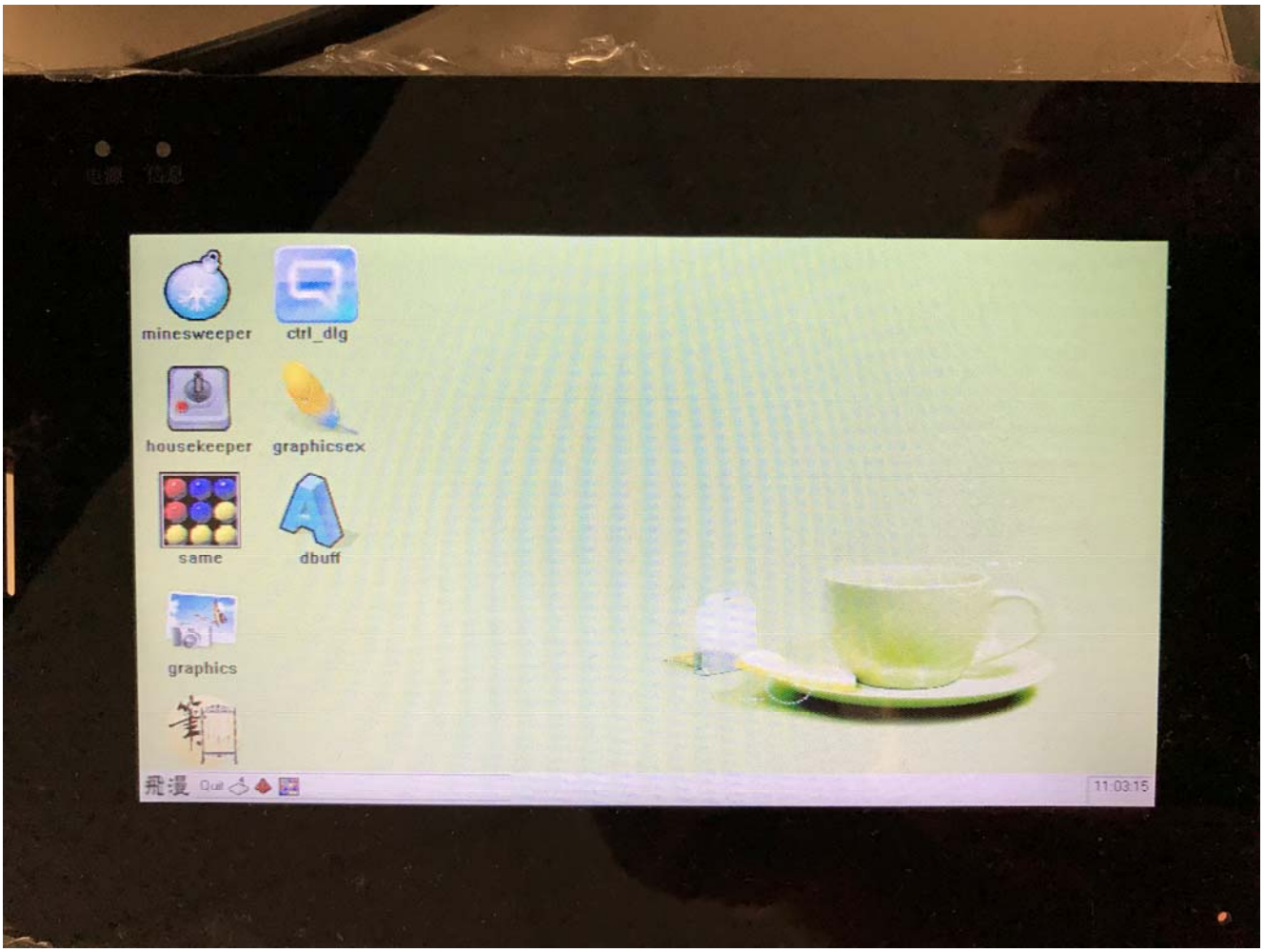
MiniGui 采用 server/client 模式工作，在运行 demo 之前需要先运行 APP 编译出来的 mginit 将 server 开启

3.2.3 demo

需要注意 MiniGui demo 读取私有资源文件的方式，采用的是相对路径。

4. DEMO 运行实例

4.1. Desktop



4.2. Housekeeper

4.3. Datepicker