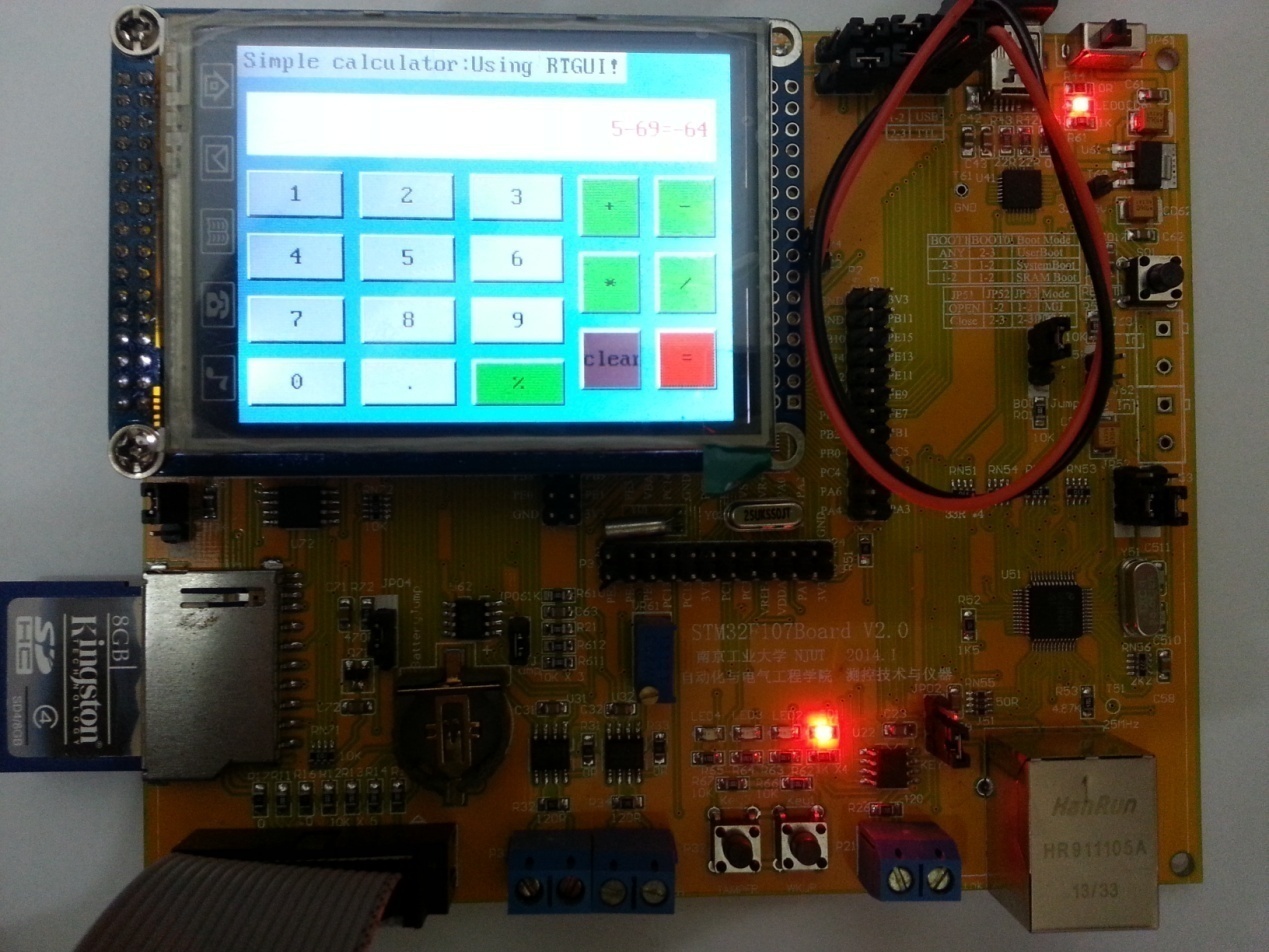
出厂时安装的固件，程序代码在STM32F107Board文件夹中。

# 1.RTGUI简介

按照jiezhi的简易计算器，在107上实现。采用了苗成诗的界面尺寸。



RTGUI的资料非常少，比emWIN 要少太多，所以使用起来非常不方便，没有官方手册，只有例程。但是由于其与RTT结合紧密，所以还是研究了一下。

能找到的RTGUI的资料如下：

rtgui简单分析.pdf

RT-Thread GUI应用开发介绍.pdf

第一章 RTGUI控件对象的管理机制 .pdf

一起来学RT-Thread教程连载-更新到20篇.pdf

# 2.LCD驱动

除了传统的LCD驱动代码，还要添加 以下代码内容：

rtgui调用的函数有：

/\* 在某个坐标上画像素点 设置像素点顔色 \*/

void rt\_hw\_lcd\_set\_pixel(const char\* c, int x, int y)

/\* 读取某个坐标上像素点，按照RGB格式返回\*/

void rt\_hw\_lcd\_get\_pixel(char\* c, int x, int y)

/\* 画水平线 \*/

void rt\_hw\_lcd\_draw\_hline(const char\* c, int x1, int x2, int y)

/\* 画垂直线 \*/

void rt\_hw\_lcd\_draw\_vline(const char\* c, int x, int y1, int y2)

/\* 将某个数组中的数据copy至lcd的gram中 \*/

void rt\_hw\_lcd\_draw\_blit\_line(const char\* c, int x, int y, rt\_size\_t size)

/\* 类似于linux的设备驱动框架\*/

struct rt\_device\_graphic\_ops lcd\_ili\_ops =

{

rt\_hw\_lcd\_set\_pixel,

rt\_hw\_lcd\_get\_pixel,

rt\_hw\_lcd\_draw\_hline,

rt\_hw\_lcd\_draw\_vline,

rt\_hw\_lcd\_draw\_blit\_line

};

static rt\_err\_t lcd\_init(rt\_device\_t dev)

static rt\_err\_t lcd\_open(rt\_device\_t dev, rt\_uint16\_t oflag)

static rt\_err\_t lcd\_close(rt\_device\_t dev)

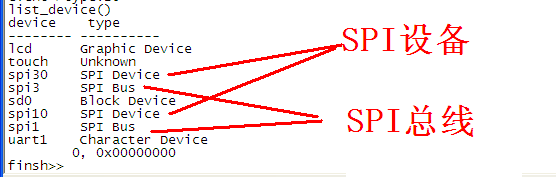
static rt\_err\_t lcd\_control(rt\_device\_t dev, rt\_uint8\_t cmd, void \*args)

/\* 需要直接调用，用于硬件初始化和注册设备\*/

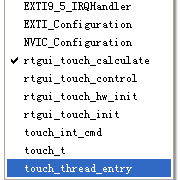
void ssd1289\_init(void)

# 3.touch驱动

触摸屏使用4线电阻式，关于原理在上面的资料已经有详细说明。采用SPI总线形式进行读写。但是在RTT中使用了SPI设备和SPI总线。



这里SPI1用于SD卡的文件系统。SPI3用于触摸屏XPT2046。

这些代码 都要重写。

下面重点讲一下初始化流程。

## 1 platform.c

在rt\_platform\_init（）函数中

首先调用 rt\_hw\_spi3\_init() 对SPI3硬件初始化；调用rt\_spi\_bus\_attach\_device(&spi\_device, "spi30", "spi3", (void\*)&spi\_cs); 注册spi30设备

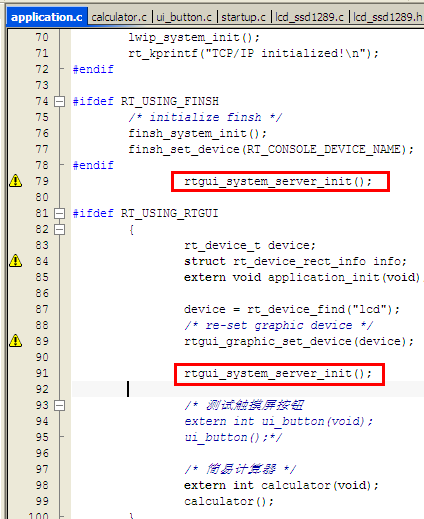
最后调用rt\_device\_init\_all 初始化所有device，包括spi30-touch设备—注意这一句非重要。

## 2. application.c

首先采用rtgui\_system\_server\_init对rtgui进行组件初始化，运行一个rtgui\_app。

然后需要运行一个ui, 则再开启一个thread, 再运行一个rtgui\_app.

非飞的解释是这样的：一个ui程序必须要有一个rtgui\_app,一个rtgui\_app是附着在一个thread上的。



## 3. 例程(一共三个)

触摸屏校准

calibration();

测试触摸屏按钮 及简易计算器

ui\_button();

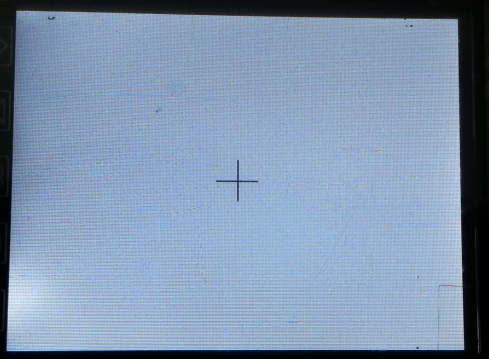
运行demo :

application\_init();

三个例子中，demo运行时需要大量内存，不能同时开LWIP.

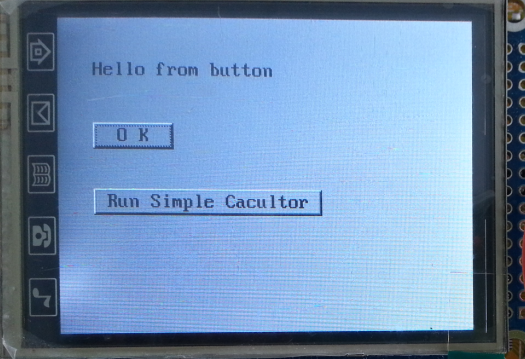
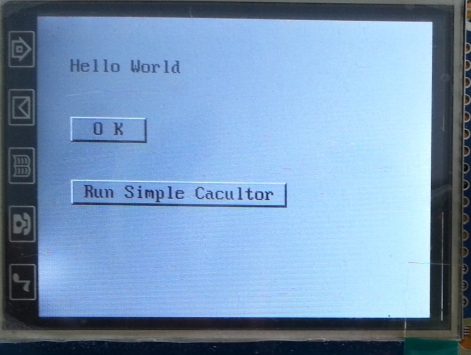
触摸屏校准

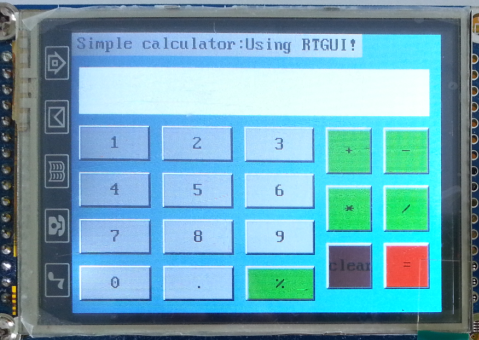
calibration();



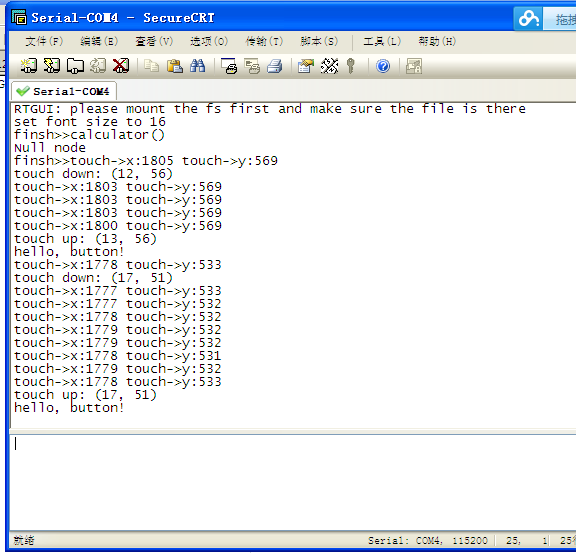
测试触摸屏按钮 及简易计算器

ui\_button();





串口助手显示:

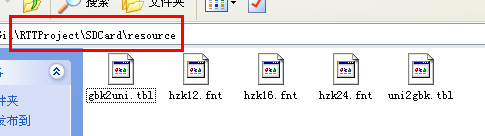


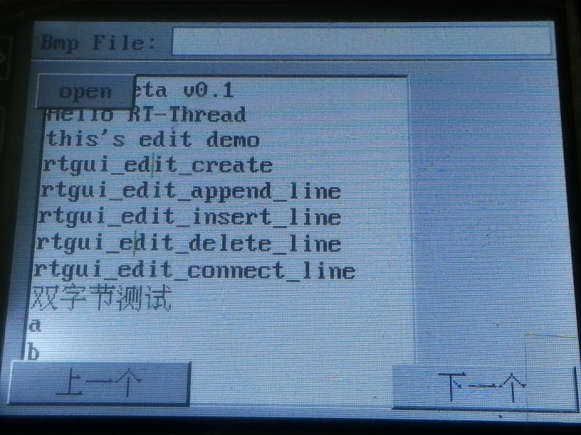
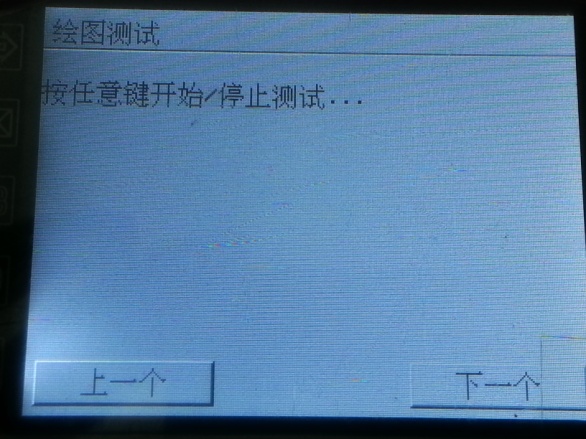
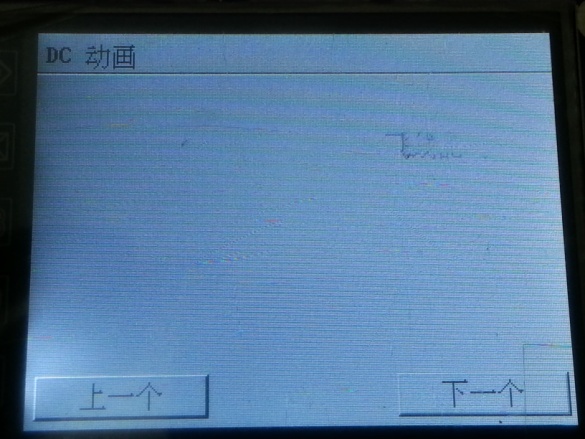
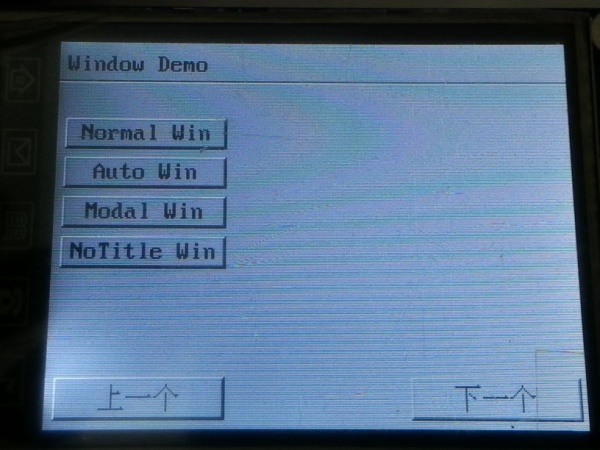
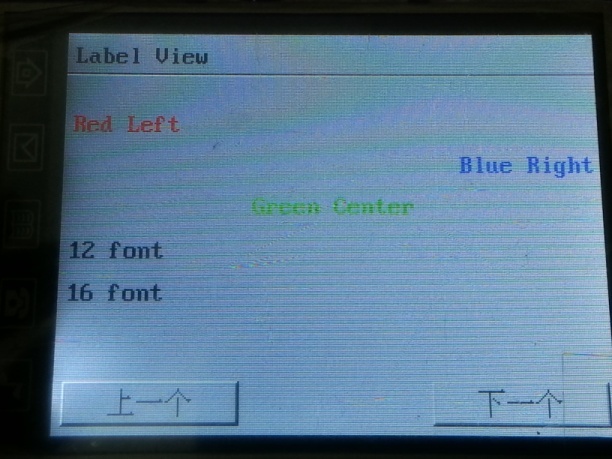
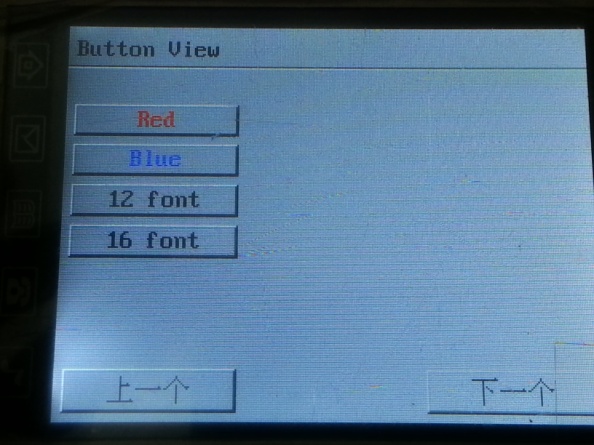
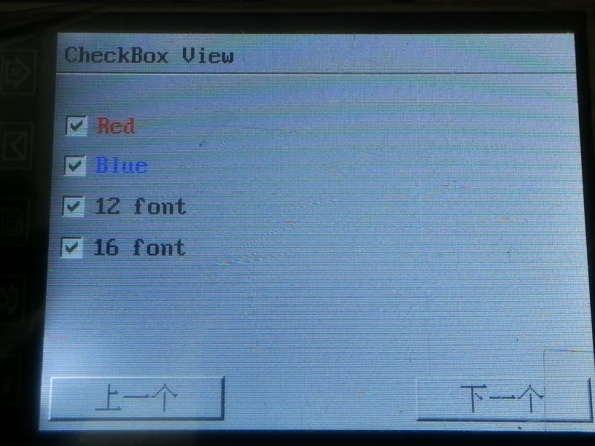
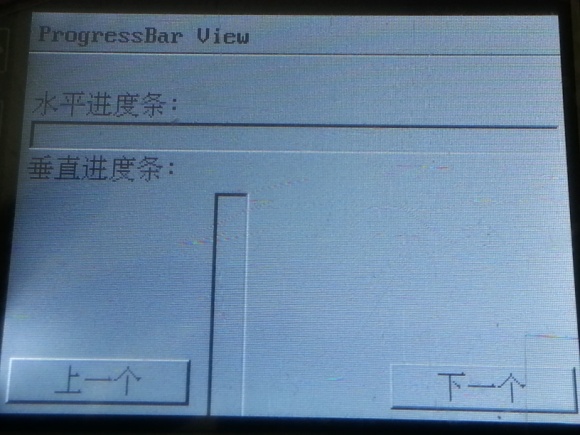
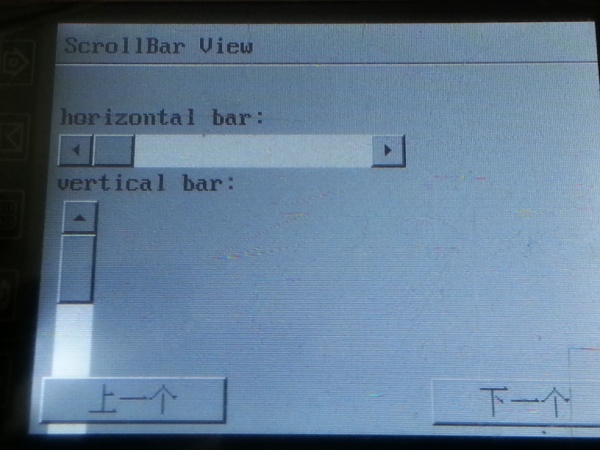
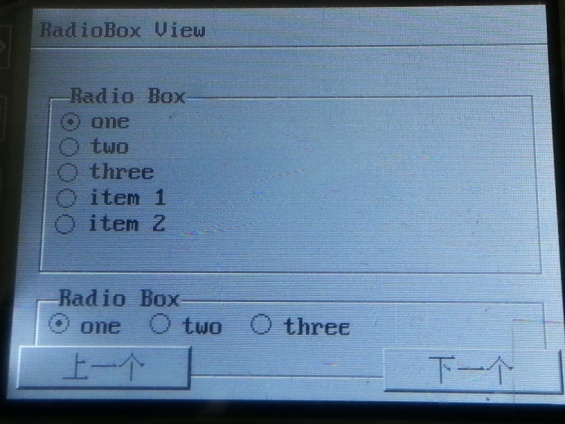
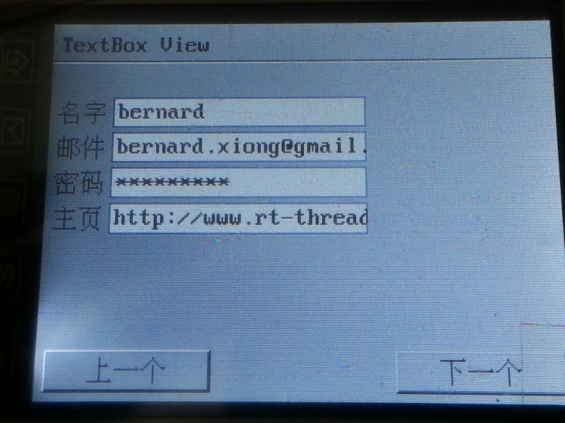
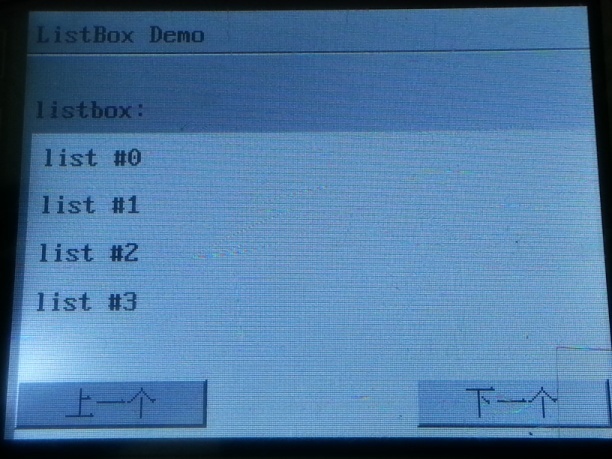
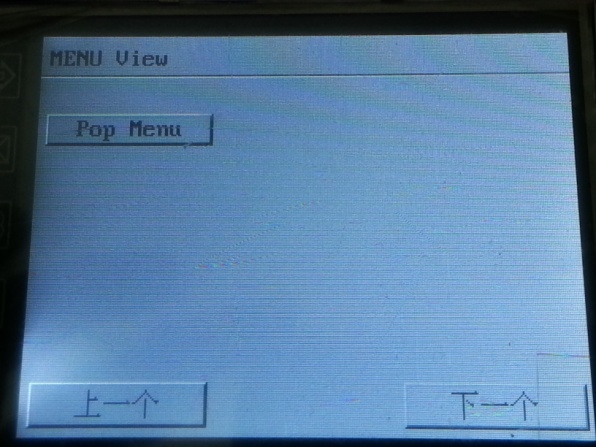
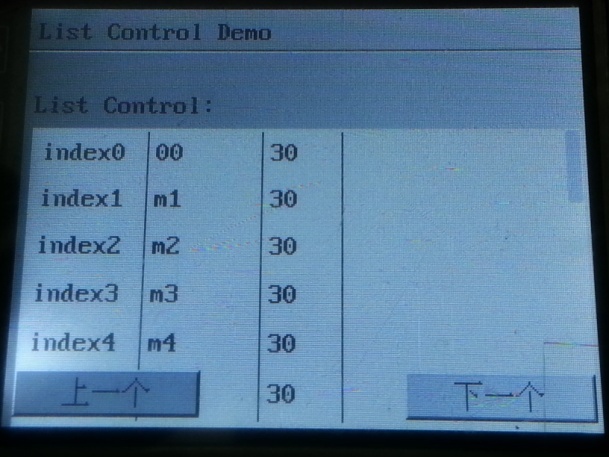
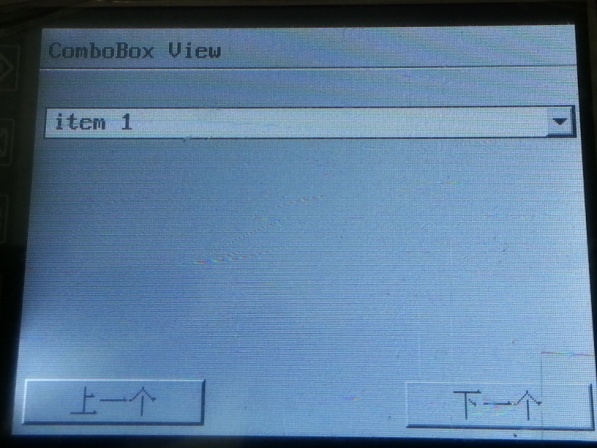
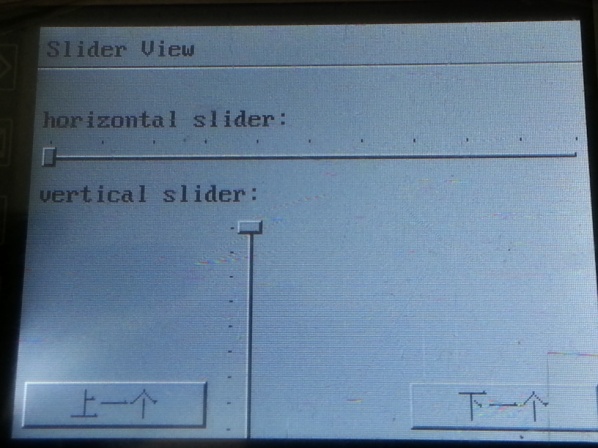
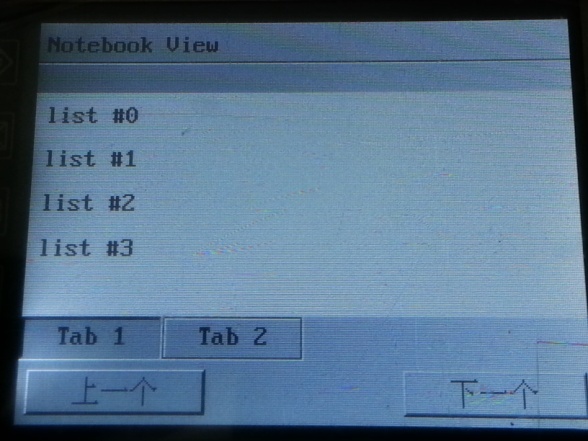
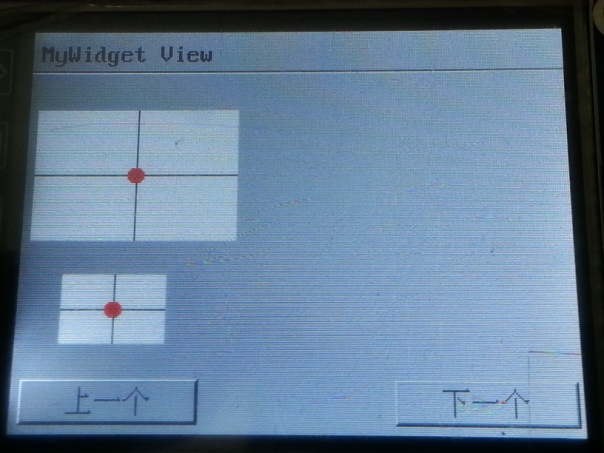
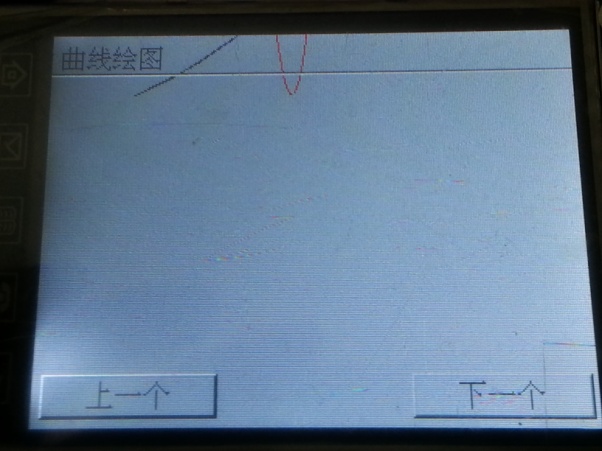
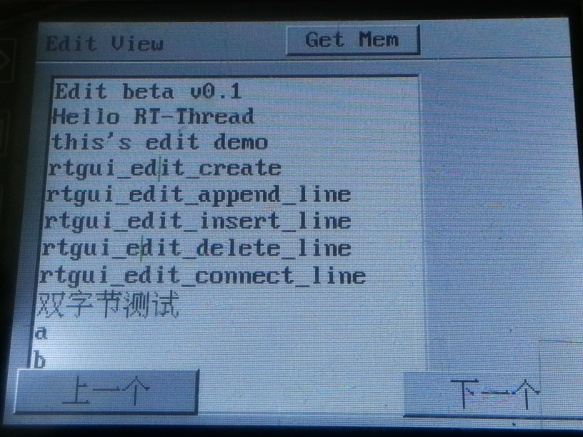
calculator();--一个简易计算器

都来自于jiezhi的例程，没有什么好说的了。

运行demo必须去除LWIP，重新生成工程

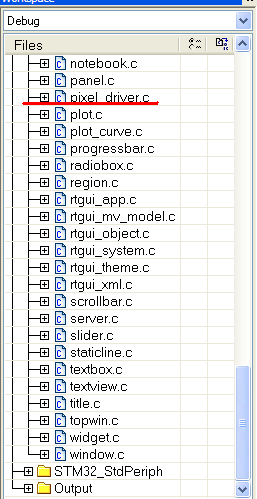
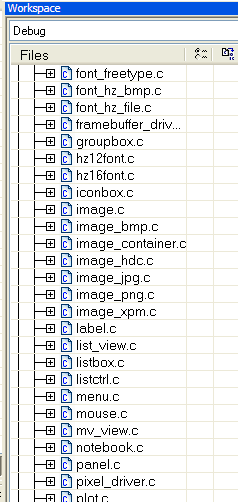
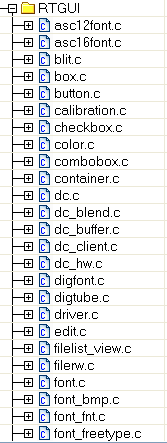
SD卡上有字库文件，在文件夹resource下，如果没有字库文件，则无法显示汉字。



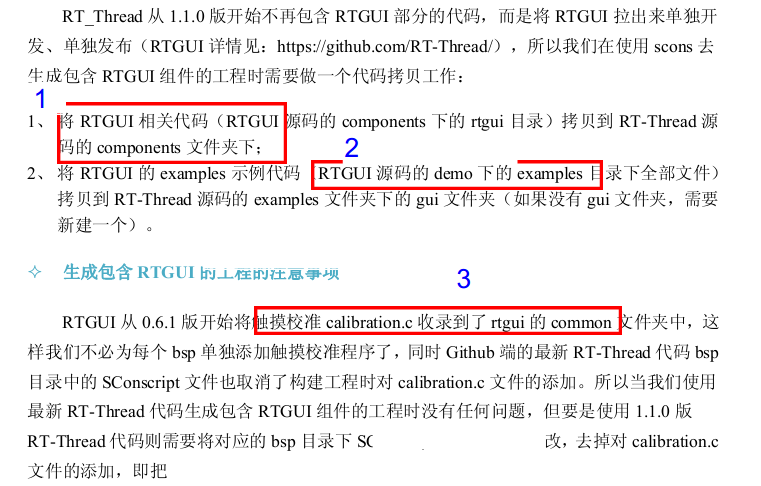


# 4. 添加RT\_USING\_RTGUI

要将相关文件加入进工程。可以使用scons重新构建工程。如不会使用，参考以下目录进行添加：



按照一起来学RT-Thread教程连载-更新到20篇.pdf P73 添加代码：



# 5.rtconfig.h中添加定义

// <section name="RT\_USING\_RTGUI" description="RTGUI, a graphic user interface" default="true" >

#define RT\_USING\_RTGUI

// <integer name="RTGUI\_NAME\_MAX" description="Maximal size of RTGUI object name length" default="16" />

#define RTGUI\_NAME\_MAX 16

// <bool name="RTGUI\_USING\_FONT16" description="Support 16 weight font" default="true" />

#define RTGUI\_USING\_FONT16

#define RTGUI\_DEFAULT\_FONT\_SIZE 16

// <bool name="RTGUI\_USING\_FONT12" description="Support 12 weight font" default="false" />

//#define RTGUI\_USING\_FONT12

// <bool name="RTGUI\_USING\_FONTHZ" description="Support Chinese font" default="false" />

#define RTGUI\_USING\_FONTHZ

// <bool name="RTGUI\_USING\_DFS\_FILERW" description="Using DFS as file interface " default="true" />

#define RTGUI\_USING\_DFS\_FILERW

// <bool name="RTGUI\_USING\_HZ\_FILE" description="Using font file as Chinese font" default="false" />

#define RTGUI\_USING\_HZ\_FILE

// <bool name="RTGUI\_USING\_HZ\_BMP" description="Using Chinese bitmap font" default="false" />

//#define RTGUI\_USING\_HZ\_BMP

// <bool name="RTGUI\_USING\_SMALL\_SIZE" description="Using small size in RTGUI" default="false" />

// #define RTGUI\_USING\_SMALL\_SIZE

// <bool name="RTGUI\_USING\_MOUSE\_CURSOR" description="Using mouse cursor in RTGUI" default="false" />

// #define RTGUI\_USING\_MOUSE\_CURSOR

// <bool name="RTGUI\_IMAGE\_XPM" description="Using xpm image in RTGUI" default="true" />

#define RTGUI\_IMAGE\_XPM

// <bool name="RTGUI\_IMAGE\_JPEG" description="Using jpeg image in RTGUI" default="true" />

//#define RTGUI\_IMAGE\_JPEG

// <bool name="RTGUI\_IMAGE\_PNG" description="Using png image in RTGUI" default="false" />

//#define RTGUI\_IMAGE\_PNG

// <bool name="RTGUI\_IMAGE\_BMP" description="Using bmp image in RTGUI" default="true" />

#define RTGUI\_IMAGE\_BMP

// <bool name="RTGUI\_USING\_NOTEBOOK\_IMAGE" description="Using notebook image in RTGUI" default="true" />

#define RTGUI\_USING\_NOTEBOOK\_IMAGE

// <bool name="RTGUI\_USING\_HW\_CURSOR" description="Using hardware cursor in RTGUI" default="true" />

//#define RTGUI\_USING\_HW\_CURSOR