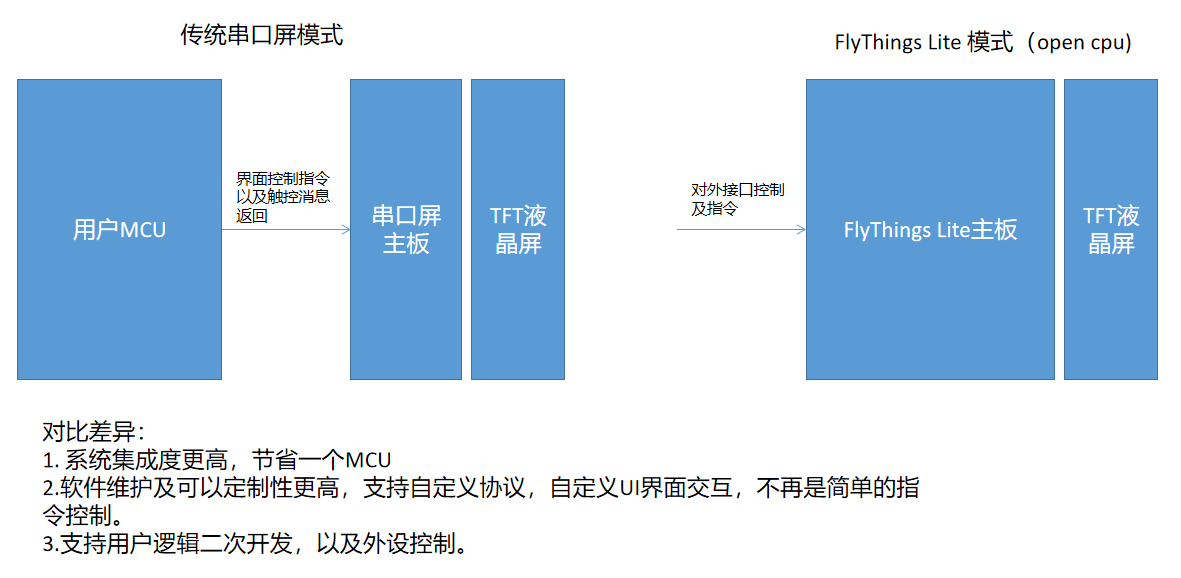
FlyThings Lite使用说明文档1.0

深圳市中科世为科技有限公司

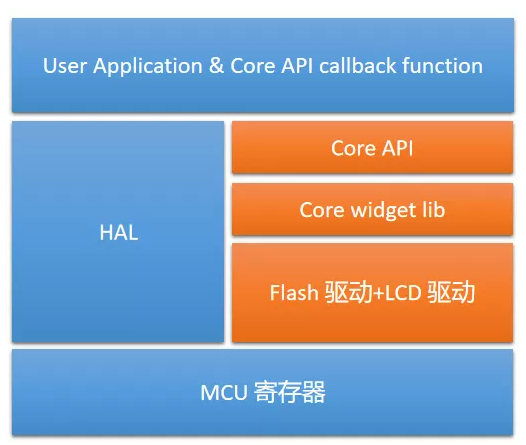
# FlyThings Lite介绍：

FlyThings Lite是中科世为基于MCU平台开发的超低成本界面系统。低于2K的RAM，16KFlash的硬件资源消耗。使用FlyThingsLite IDE开发工具快速设计界面。支持ZKBITMAP高达90%压缩率的BITMAP编解码算法。Opencpu的模式，更便于用户使用，相比传统串口屏直接节省一个MCU。

## 与传统串口屏的对比：



## 系统框架图：



其中黄色部分为FlyThings Lite核心GUI部分内容，不开放源码。

但是用户可以通过flyport部分适配不同的显示屏以及Flash的驱动。

# 环境搭建：

## 准备材料：

Windows 7或以上的电脑，32bit/64bit

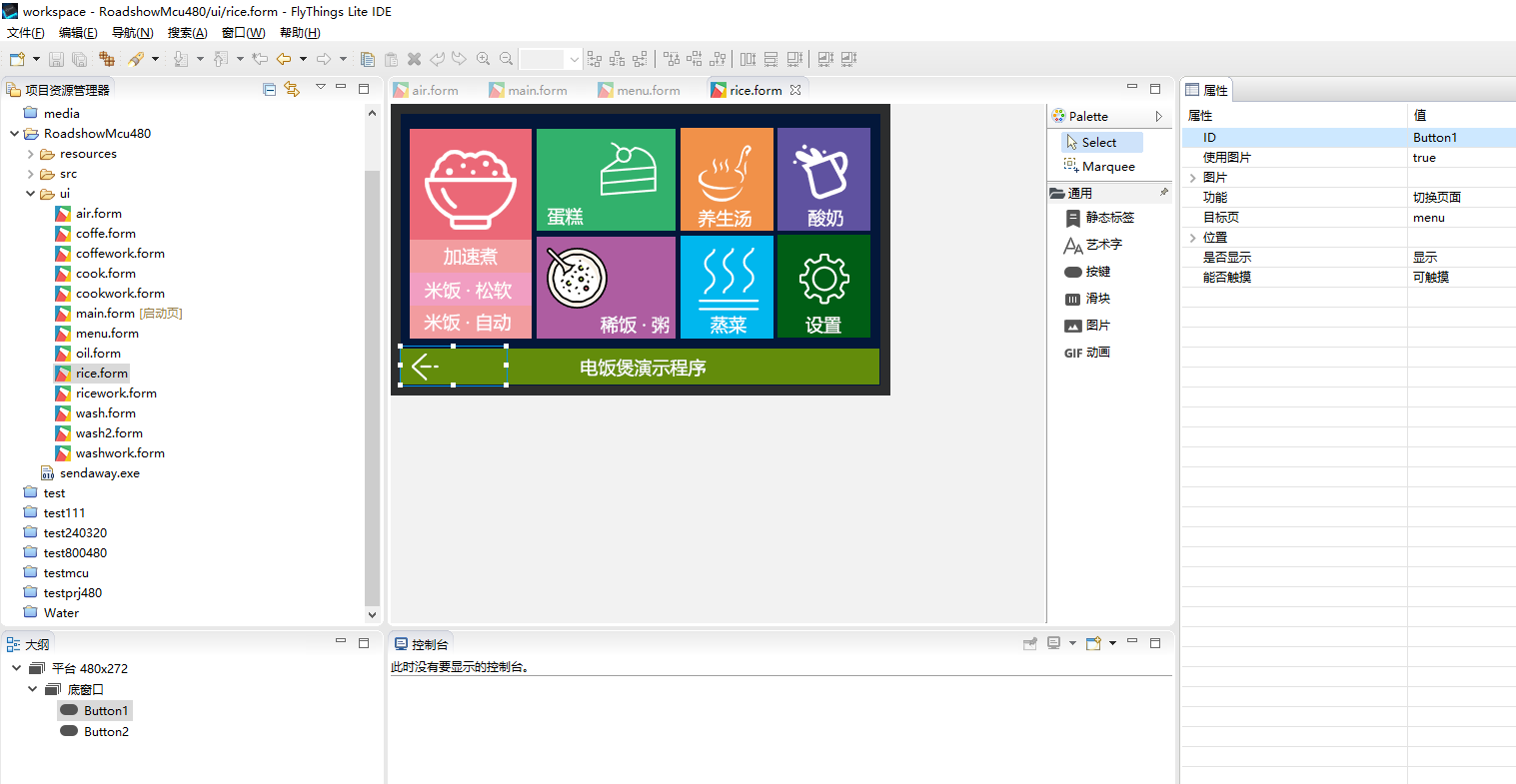
USB转TTL或者本身带串口电脑

代码开发环境（视具体MCU开发环境，比如：一般MCU使用Keil，ESP32则使用idf环境）

UI开发工具下载：

<http://www.zkswe.com/h-col-135.html> 找到FlyThings Lite 下载安装即可（安装注意路径不可以包含空格及中文）。

安装后打开的效果：



## 新建FlyThings Lite项目

找到软件顶部菜单栏，依次选择 文件 -> 新建 -> FlyThings for MCU Project

输入项目名称，选择平台。选择分辨率，或者按照格式，输入自定义的分辨率。点击完成。

在软件的左侧，项目资源管理器区域，找到新建的项目。

选中项目名，右键，在弹出的菜单中，选择模拟器运行。  
软件右下角给出正在编译的提示，几秒钟后，自动弹出模拟器，查看运行效果。

默认新建的项目，只有一个白色的窗口，参照下一章节[添加控件](https://lite.flythings.cn/zh-hans/add_widget.html)。

## 添加控件

找到项目资源管理器，双击指定的项目名，打开目录。  
双击ui文件夹下的main.form文件，将在右侧看到该UI文件的预览效果。

在预览图的右侧控件列表里，找到静态标签控件，  
鼠标左键点击静态标签控件不松开，然后将其拖拽到预览图上的空白位置，松开左键，就能看到自动生成的控件。

在软件的左下角，有一个大纲视图，可以查看底窗口下包含的所有控件。

在预览图中，左键点击刚才添加的控件，软件的最右侧属性栏，将显示该控件的所有可编辑的属性。

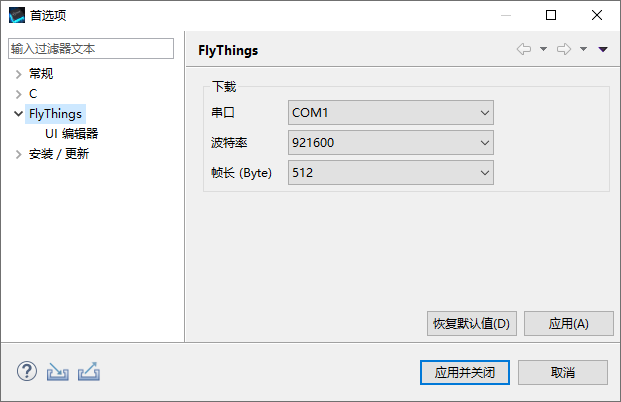
在属性栏里，可以自由编辑，大部分的属性，在修改后，可以立即看到预览效果。  
或者选择模拟器运行，查看完整效果。

## 下载程序到MCU

保持电脑与MCU串口正常连接。

找到软件顶部的菜单栏，依次选择窗口->首选项->FlyThings Lite  
在下载选项组中，选择与MCU连接的串口。  
尽量选择MCU能支持的最高波特率，这将有效缩短下载时间。  
其他保持默认。

设置完成后，点击应用并关闭保存设置。



在项目资源管理器中，选中想要下载的项目名，右键，在弹出的菜单中，选择下载到MCU。  
软件右下角提示正在编译，几秒钟后，自动弹出下载程序，如果一切正常，将看到下载进度。  
如果下载程序提示超时，请确认串口连接情况，以及MCU是否正常开机。

## UI界面开发

细节参考：<https://lite.flythings.cn/zh-hans/label.html>

# 功能二次开发

## 代码下载：

<https://gitee.com/Kwolve/flythings-lite-release>

找到对应MCU的工程文件打开

## MCU软件移植说明：

将FlyThings Lite开发工具工程目录下src目录的文件拷贝到MCU源码的uifun目录下，并在keil下添加对应的c源码文件即可。

### 软件接口：

src/logic目录下的问价每个界面都会对应一个c文件如main.c

**#include** "logic/main.h"

**void** **ui\_main\_on\_page\_open**(uint8\_t page) {

页面打开的时候回调用这个页面，其中page是页面id号。

}

**void** **ui\_main\_on\_timer**(uint32\_t millisecond, uint32\_t count) {

在flyport/timer.c中适配的定时器运行的时候这边会产生回调。其中millisecond是间隔时间，count是回调次数。也就是实际工作时间=millisecond\*count

}

**void** **ui\_main\_on\_button\_touch\_event**(button\_t \*button, touch\_type\_e type) {

触控按键的时候调用，button是按键对应的控件名，可以通过logic/main.h 中的宏定义匹配按键。

如下是main.h 中的内容。

**#define** UI\_WIDGET\_MAIN\_Animation1 ((animation\_t\*)gui\_context\_find\_widget(2))

**#define** UI\_WIDGET\_MAIN\_Button1 ((button\_t\*)gui\_context\_find\_widget(3))

代码可以描述为：

if(button == UI\_WIDGET\_MAIN\_Button1)

来指示按键按下的是button1.

}

**void** **ui\_main\_on\_progress\_changed**(progress\_t \*progress, uint16\_t val) {

页面进度条进度通知，其中progress同上的button的说明，val是进度值。

}

**void** **ui\_main\_on\_animation\_play\_end**(animation\_t \*animation) {

动画播放结束的响应代码，animation同上的button的说明。

}

## 控件描述：

### Label 控件： 无操作api

### Button控件：

void button\_switch(button\_t \*button, bool on);

设置按键的开关状态，button是按键的id，通过页面的宏定义设置，如上的：UI\_WIDGET\_MAIN\_Button1

参考调用方法：

button\_switch(UI\_WIDGET\_MAIN\_Button1, true);

### 艺术字：

void imagetext\_set\_text(imagetext\_t \*imagetext, const char \*text);

设置艺术字显示的内容，imagetext同上介绍，为UI控件。Text是字符串内容。

调用方法：

imagetext\_set\_text(UI\_WIDGET\_MAIN\_IMAGETEXT1,”12:00”);

### 进度条

void progress\_set\_value(progress\_t \*progress, uint16\_t val);

设置进度条的进度值，progress同上介绍，为UI控件。Val是进度条进度值。

调用方法：

progress\_set\_value(UI\_WIDGET\_MAIN\_PROGRESS1, 50);

### 动画

void animation\_start(animation\_t \*anim);

开始动画播放

void animation\_stop(animation\_t \*anim);

结束动画播放

void animation\_reset(animation\_t \*anim);

动画从头开始播放

### 图片：

void widget\_set\_bg\_img(widget\_t \*widget, uint16\_t imgid);

设置图片背景图，widget为控件名称，imgid为图片的id号，id号参见工程目录下的ui\_res\_image.h的定义。