
| | | | |
|---------------|------------------|--------|----------|
| 深圳市启明智显科技有限公司 | 文 档 编 号 | 版本号 | 密级 |
| | | V1.0.0 | 无 |
| 文档名称 | 8ms 平台编译 86 盒说明书 | | 日期 |
| | | | 2022-3-8 |

8ms 平台编译 86 盒说明书



深圳市启明智显科技有限公司 版权所有



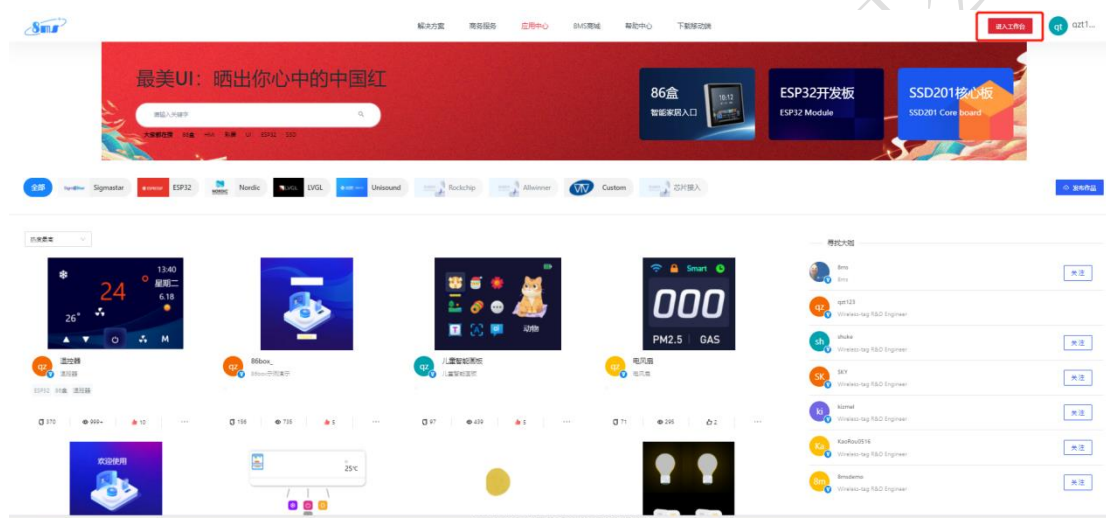
8ms 功能概括

一、创建工程

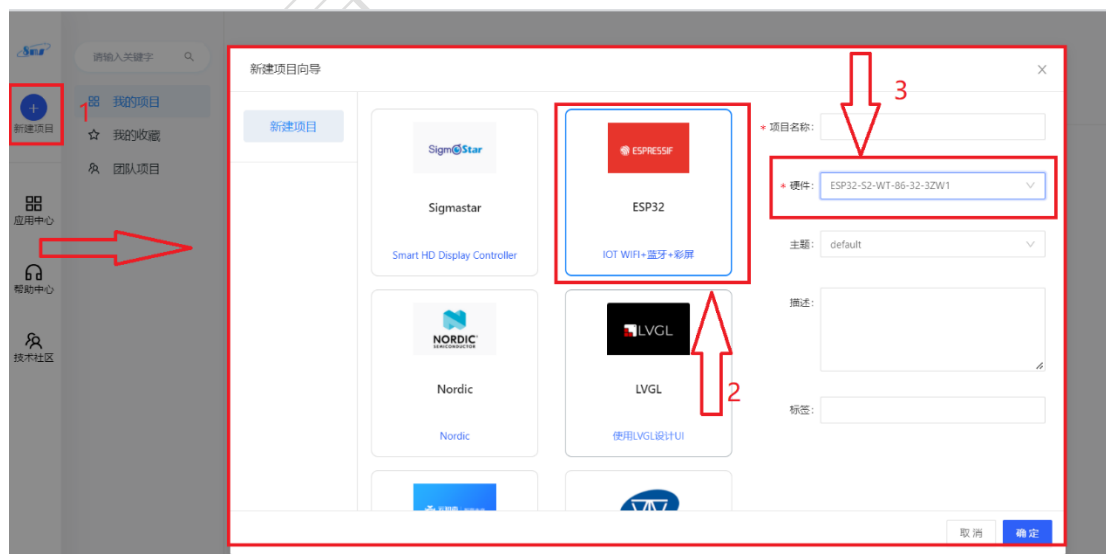
创建项目有两种方法，分别如下：

1、从空白项目创建

1.1、账号登陆成功后会直接进入应用中心，点击右上角的“进入工作台”。



1.2、点击新建项目，会弹出新建向导：



a) 选择芯片类型，输入项目名称，选择相应的硬件型号，也可添加描述和标签，点击确定即可完成新建

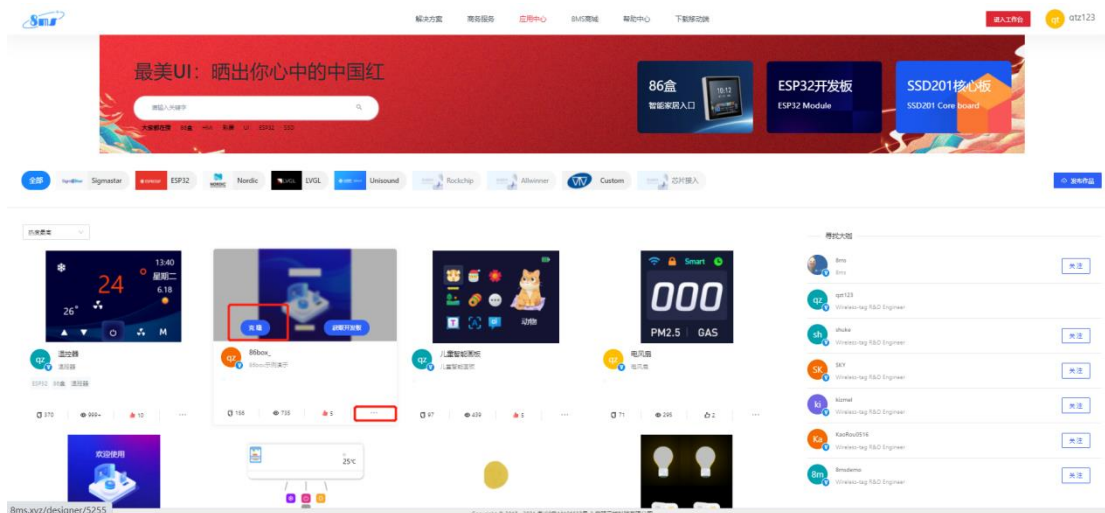
b) 项目新建好后，可在我的项目中进入项目工作区。目前硬件主要关系到不同的屏幕尺寸，



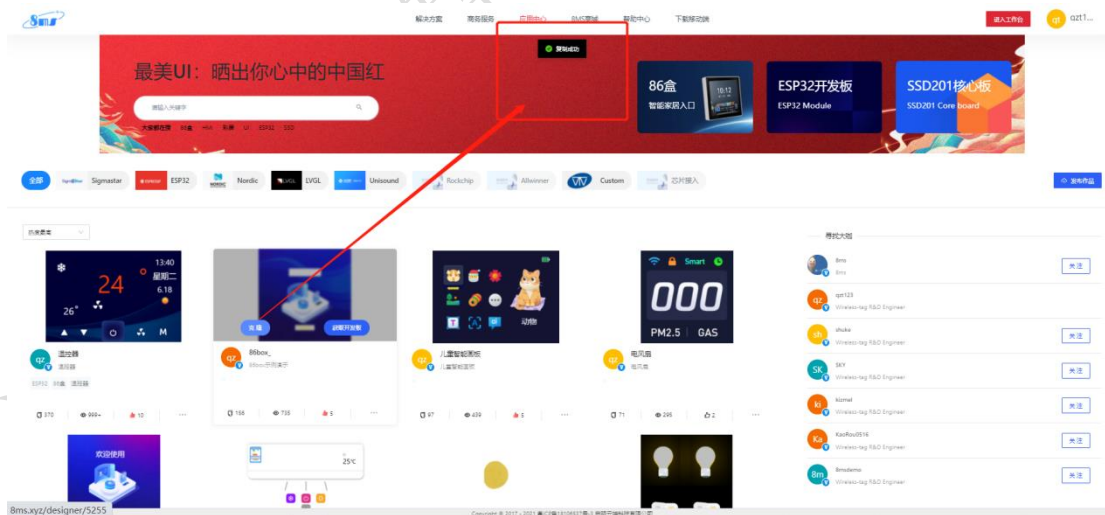
所以不同的硬件对应的画布尺寸也是不一样的

2、从推荐项目克隆

2.1、点击推荐项目里的克隆图标，可把推荐项目里的项目克隆到我的项目里。



2.2、点击克隆图标之后，会弹出复制成功字样，表示克隆成功，可在我的项目中查看。



2.3、把项目克隆下来后，可在自己的工作台对项目进行修改或者预览，也可下载源码在工程中修改，前提是需要安装 esp-idf 和下载好 86 盒的工程模板，模板工程下载地址：<https://github.com/wireless-tag-com/8ms-esp32> 下载 main 分支即可

- ①：可在线编译并且预览操作
- ②：如果对克隆项目修改，编译后可下载源码



根据下载 SDK 的 readme 提示，在控制台运行脚本 `load.sh`-->选择开发板对应的型号-->回车执行脚本替换。

选择开发板

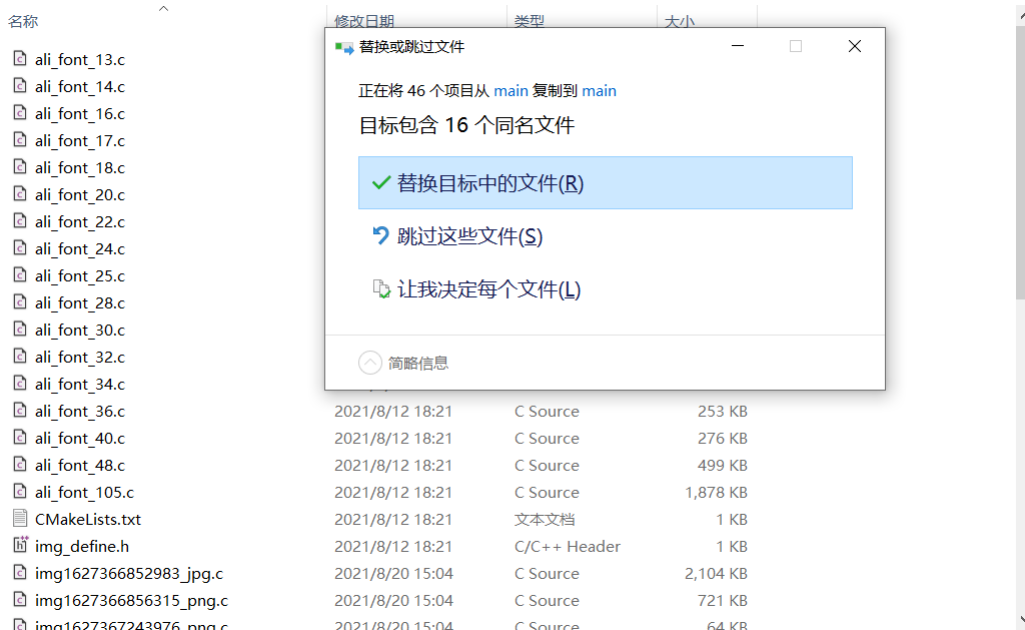
使用 `load.sh` 加载开发板配置

```
bash load.sh
```

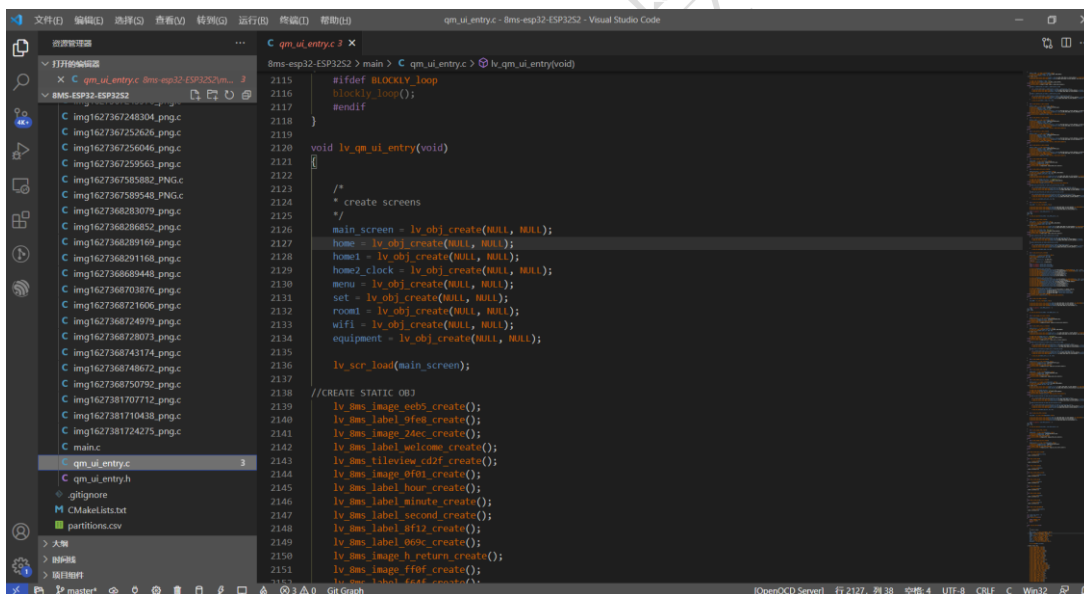
当前支持型号

1): WT32_SC01_LANDSCAPE 2): WT32_SC01_PORTRAIT 3): WT154-S2MI1-PERFORMANCE 4): WT154-S2MI1-WIFI 5): WT154_C3SI1 6): WT-86-32-3ZW0-PERFORMANCE 7): WT-86-32-3ZW0-WIFI 8): WT-86-32-3ZW1-PERFORMANCE 9): WT-86-32-3ZW1-WIFI 10): WT280-S2MT1 11): WT280-S2MI1 12): WT240-C3SI1-LANDSCAPE 13): WT240-C3SI1-PORTRAIT 14): WT-0-32-21MLW1

后面将下载的源码 `qmsd_ui` 文件夹的所有文件复制到模板 `8ms-esp32/components` 文件夹内--- 替换目标中的文件



使用 VS CODE 打开 8ms-esp32 文件夹进行修改或者直接编译下载到自己的 86 盒上



2.4、编译通过后，有两种烧录方法：

方法一：运行命令烧录：

idf.py flash -p [端口号] monitor

相关烧录命令以及过程可参考乐鑫官方文档：

https://docs.espressif.com/projects/esp-idf/zh_CN/latest/esp32/get-started/windows-setup.html#id14

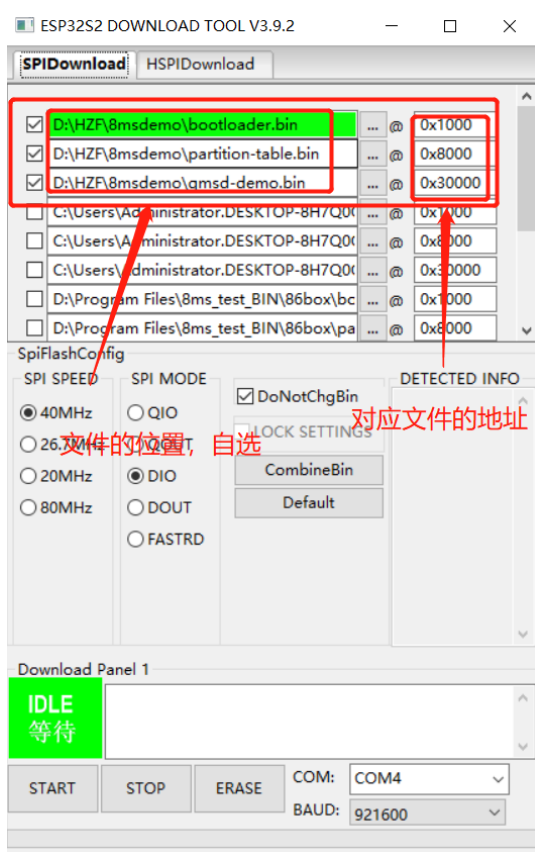
方法二：通过烧录工具烧录：

根据编译日志提示，在工程 build 目录下找到三个烧录文件

```
mode dio --flash_size detect --flash_freq 80m 0x1000 build/bootloader/bootloader.bin 0x8000 build/partition_table/partition-table.bin 0x30000 build/qmsd-demo.bin
```



将文件与烧录地址对应，使用烧录工具进行烧录：



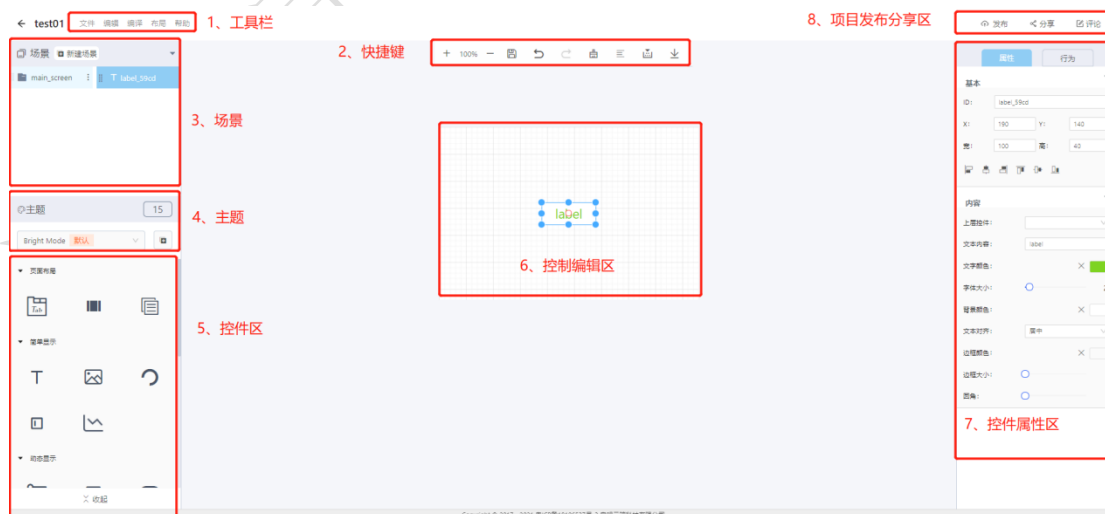
其他型号的开发板操作与 86 盒类似。需要注意的是先执行脚本后，再将下载的源码替换到工程。

在执行脚本文件 load.sh 时，需要选择与自己开发板一致的型号，否则编译有可能报错。

当编译发生错误时，须检查 idf 的版本是否时 4.3.1 以及 8ms 的 sdk 版本是否为最新版本。将错误的版本更新后，再次编译。

二、页面设计

1、工作区说明：



① 菜单栏：按照平台所支持的功能分组排列。

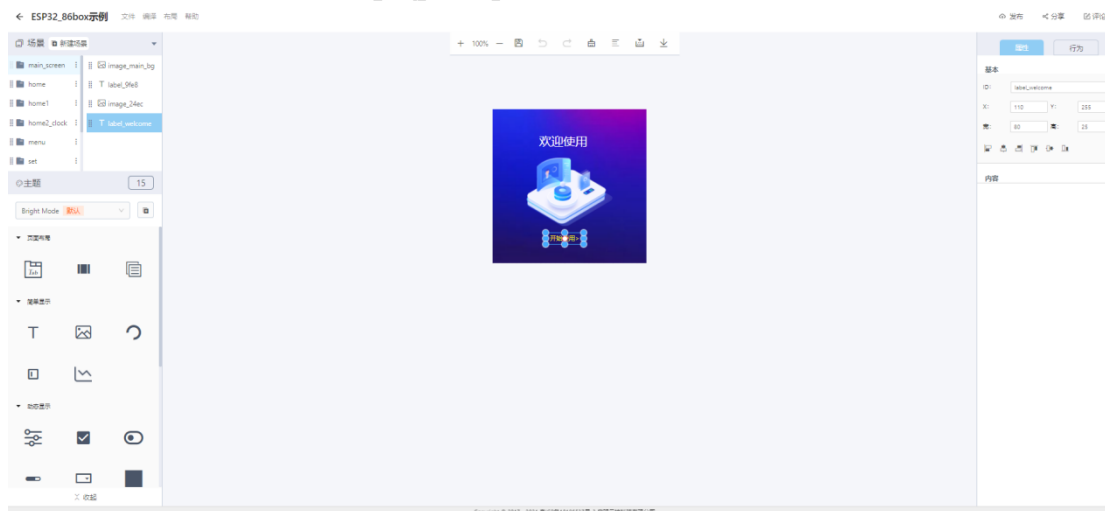


- ② 项目场景和控件列表区：列出该项目所拥有的项目场景及当前场景下的所有控件。
- ③ 控件选择区：目前平台有二十几种控件，平台所支持的所有控件（鼠标停留在控件处即可显示控件名称），点击控件即可在当前场景创建所选控件。
- ④ 快捷选择区：提供常用的快捷按钮，例如：编译、下载 bin 等功能。
- ⑤ 控件编辑区：画布大小即对应硬件平台的显示大小，控件在画布的相对位置就是对应硬件平台的显示位置。
- ⑥ 帐号项目分享、评论区：提供项目发布、分享、评论功能。
- ⑦ 场景（或控件）属性区：选择场景（或控件）后，属性区出现对应的属性设置选择。

2、 示例讲解（以 ESP32 彩色 86 盒为例可直接在应用中心克隆修改或编译下载，仅供参考）：

本示例控件只用到了 8 个，分别是“文本控件”、“图片控件”、“按钮控件”、“开关控件”、“列表控件”、“滑动面板控件”、“折线控件”、“滑块控件”，控件的使用可参考 8ms 文档（<http://doc.8ms.xyz/docs/gui/gui-1bv10ne2i2u7q>）。本示例 demo 设计了 8 个界面，分别为，如下图所示：

2.1、欢迎使用界面制作（首界面）



本界面点击“开始使用”进入“home”界面，主要使用到两个图片控件和两个文本控件。

图片控件可自适应图片大小，可先按需求设计好图片大小再上传添加。

文本控件可按需求设置大小和颜色等属性并添加点击行为。

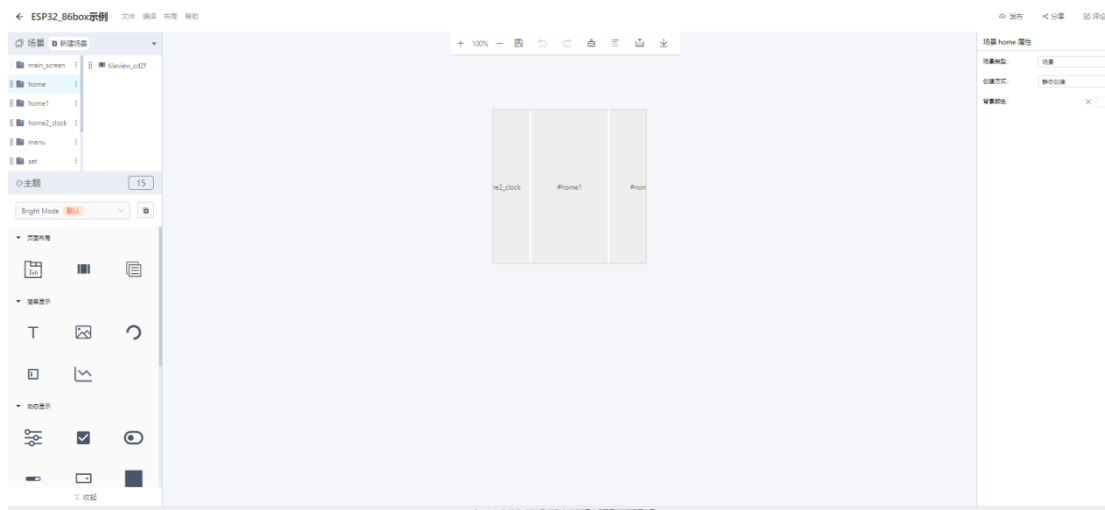


2.2、home 系列界面制作

home 系列界面制作主要用到三个场景，分别是 “home”、“home1” 和 “home2_clock”。

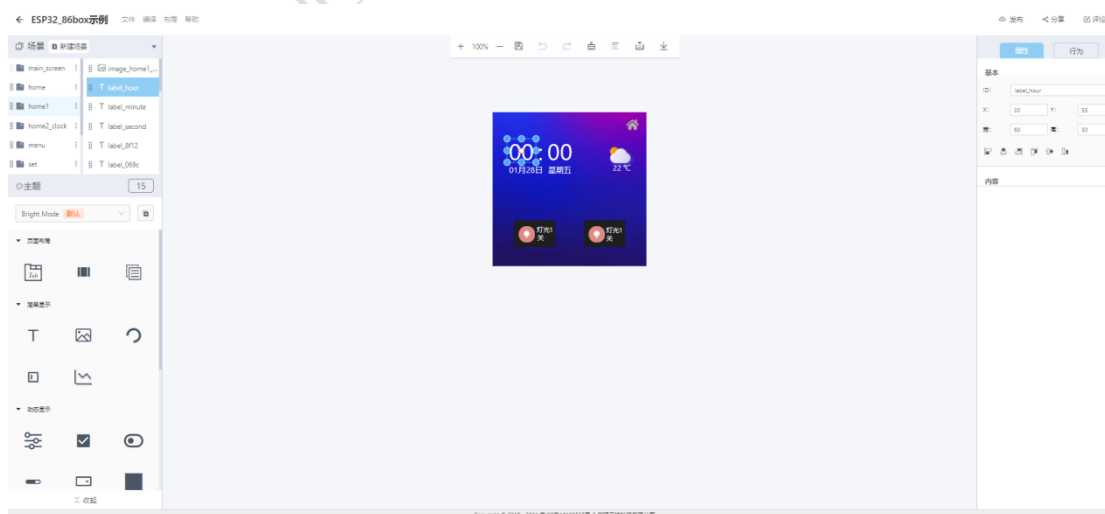
2.2.1、home 界面制作

“home” 界面使用到了“滑动面板控件”控件，该控件功能可使界面实现滑动功能，左边场景添加了 home2_clock，中间场景添加了 home1，如下图所示。



2.2.2、home1 界面制作

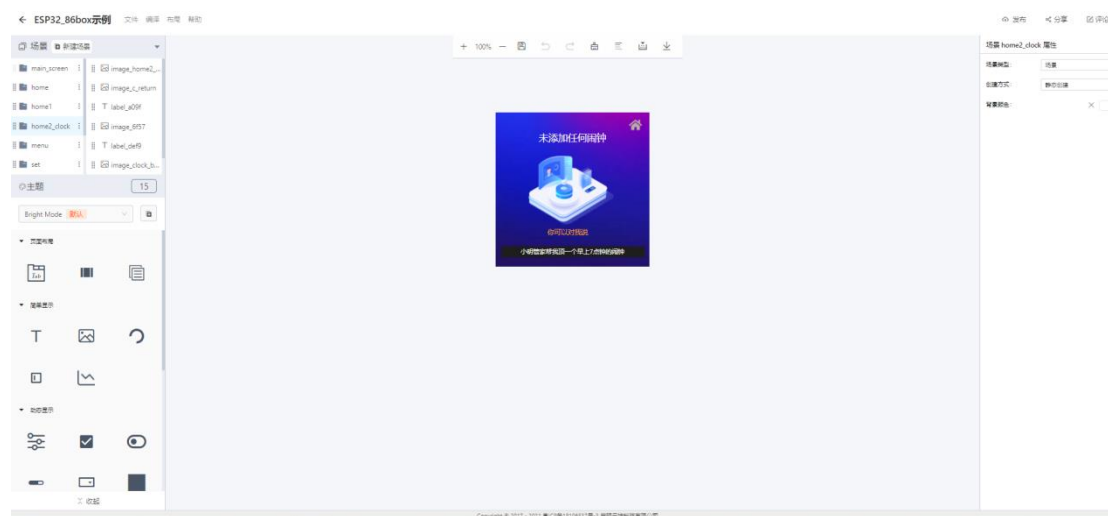
“home1” 界面主要实现两个 LED 灯亮灭的控制，时间显示和进入菜单界面。





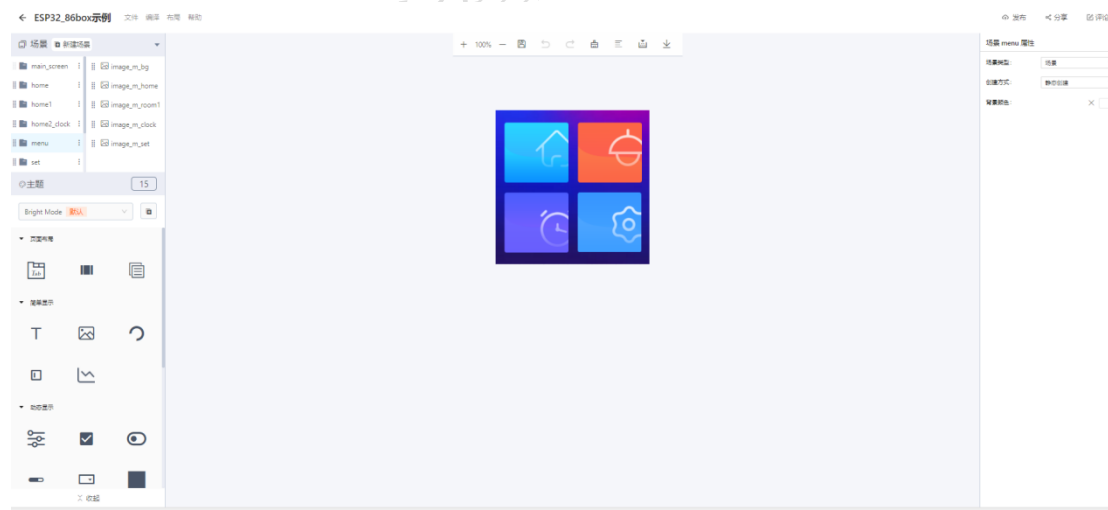
2.2.3、home2_clock 界面制作

添加闹钟界面



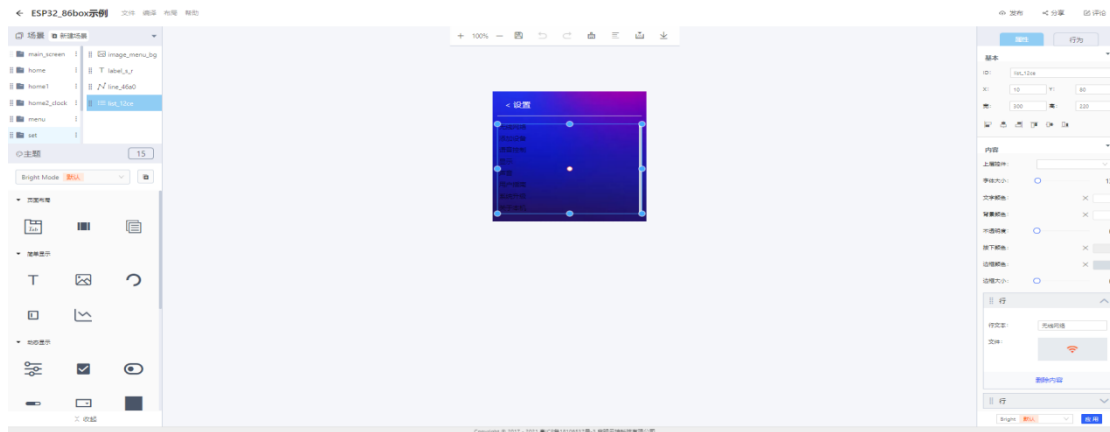
3.1、菜单界面制作

菜单界面使用了图片控件，设置了四个可点击图片跳转选项，分别是首页，房间 1，闹钟和设置。



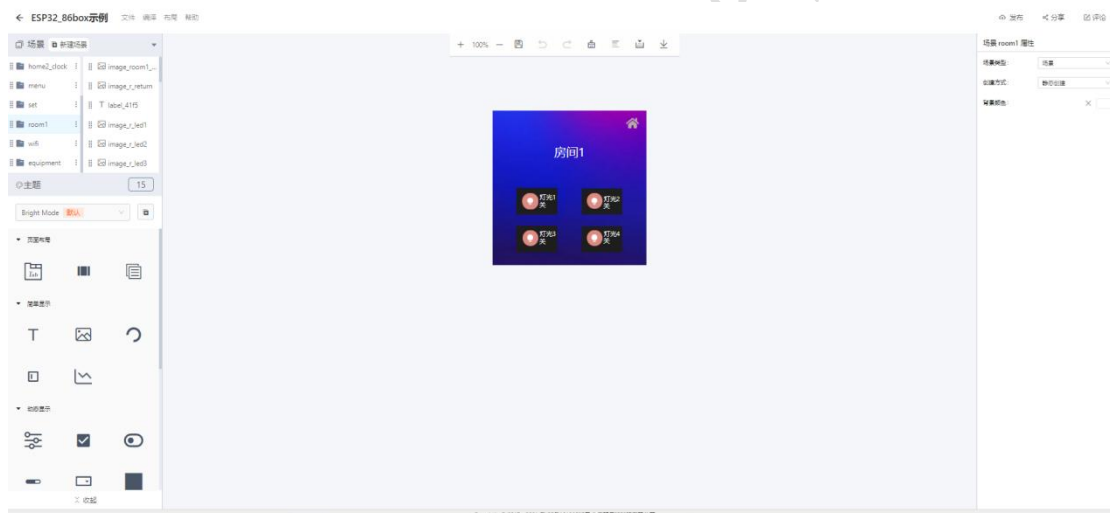
4.1、设置界面制作

设置界面，本界面使用了，label、折线和列表控件，列表控件可通过行来添加文本内容和图片。



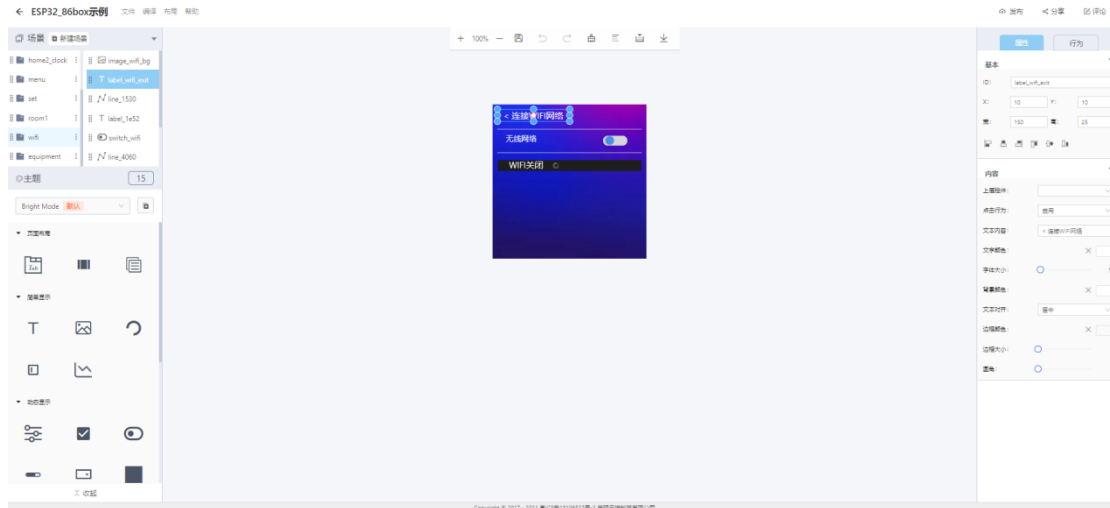
5.1、房间 1 界面制作

房间 1 界面使用到了 label 和图片控件，主要实现 4 个 LED 灯的亮灭控制。



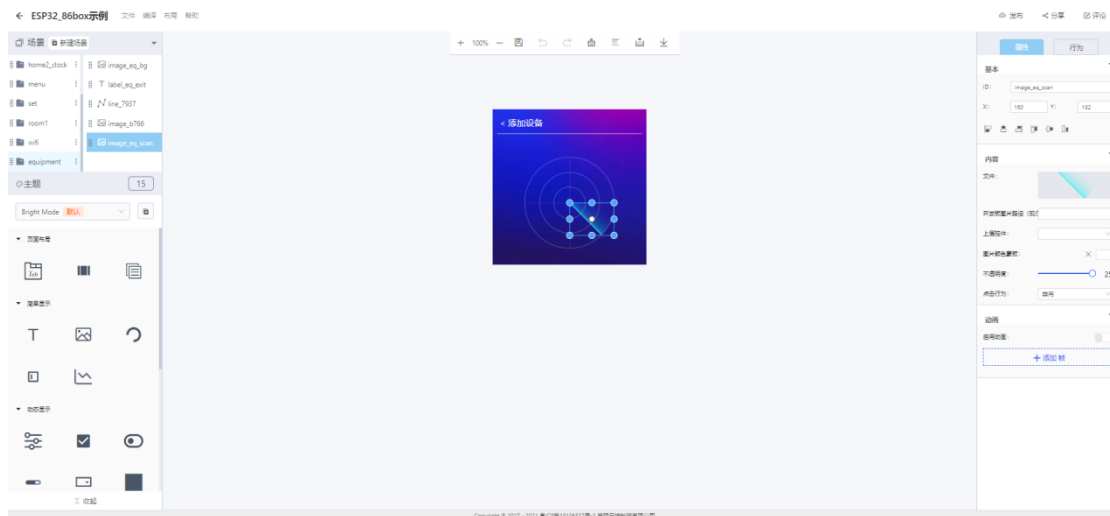
6.1、wifi 连接界面制作

wifi 界面使用到了 label、图片、折线和开关控件，通过开关控制搜索附近的 wifi 热点，搜索期间图片不断旋转。



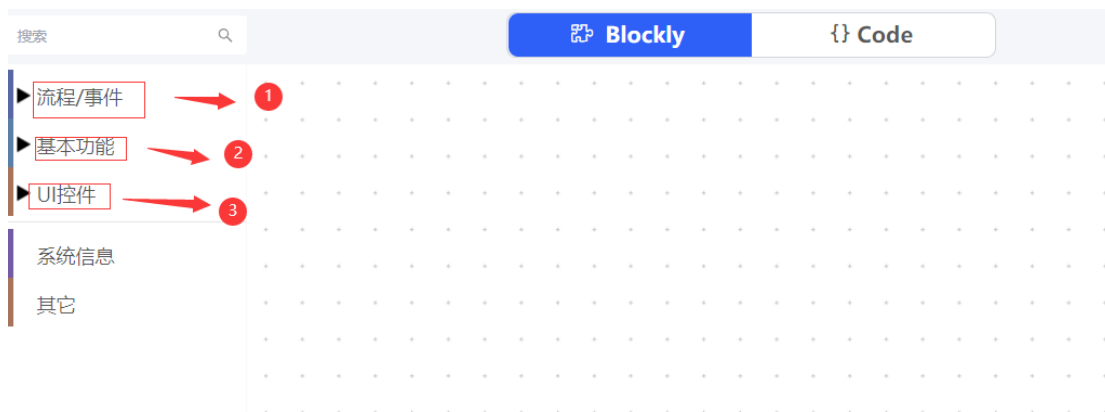
7.1、添加设备界面制作

添加设备界面使用了 label、图片和折线控件，主要实现搜索期间图片不断旋转。



三、逻辑编辑

1、 积木页面工作区分布

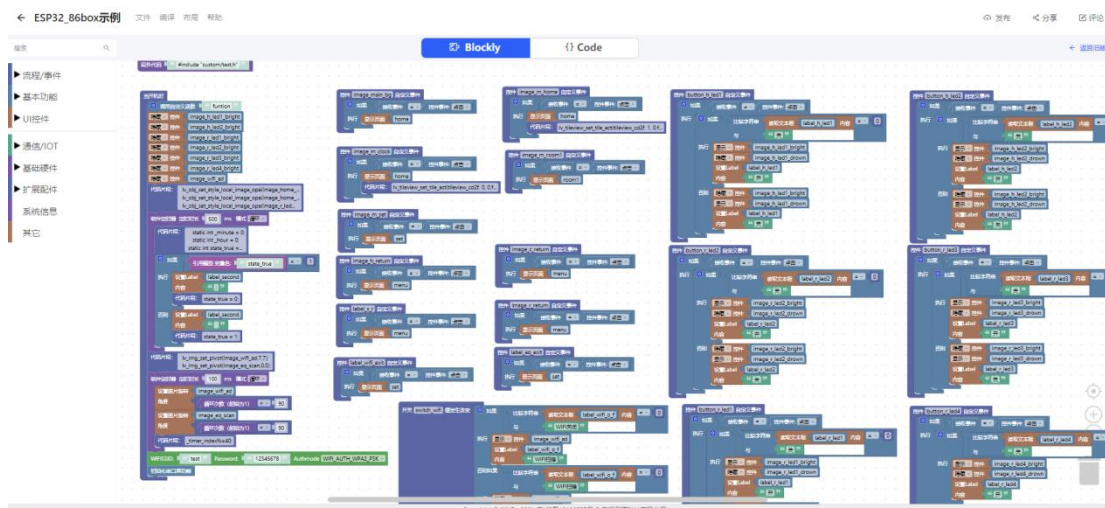


积木页面工作区主要有五大类（以 ESP32 为例），分别：

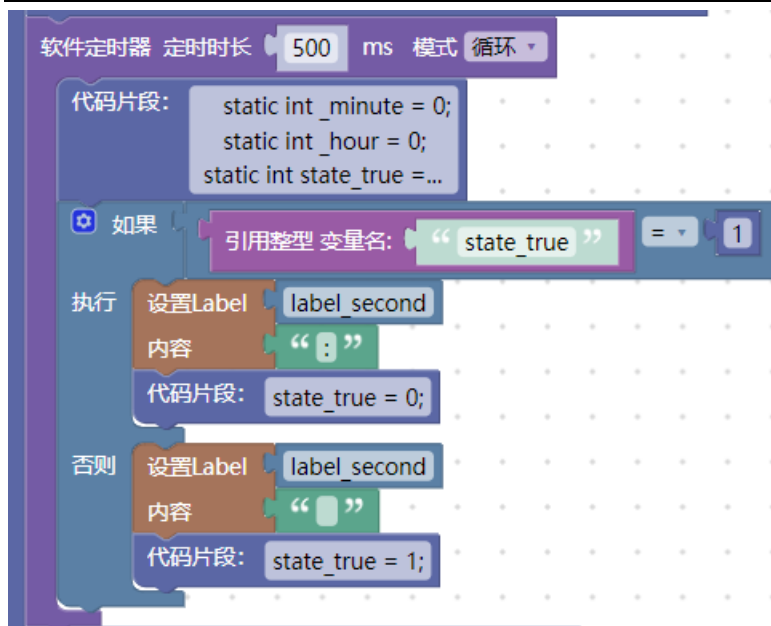
- ① 流程/事件，主要存放定时器、UI 事件和代码片段等积木块。
- ② 基本功能，主要存放 if、for 逻辑、数学运算和自定义代码等积木块。
- ③ UI 控件，主要存放对控件和场景操作的积木块，里面都对不同的控件进行分类存放。

说明：可通过点击 Code 查看已搭建的代码，详细内容可参考 8ms 积木文档（<http://doc.8ms.xyz/docs/gui/gui-1by0v24m8j8t7>）。

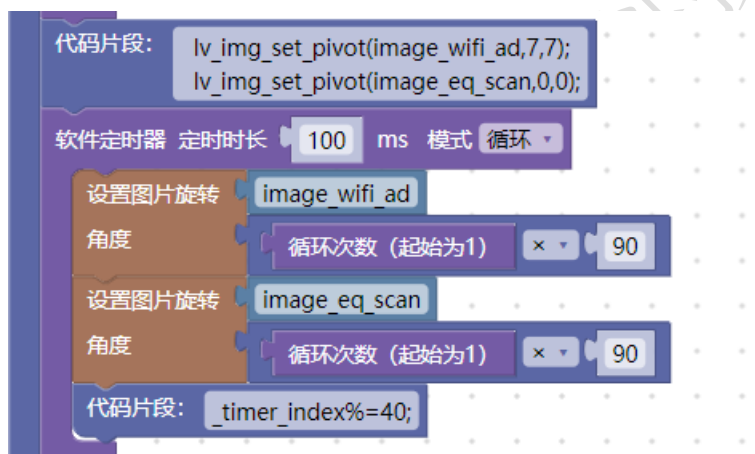
2、积木逻辑搭建，通过积木进行逻辑搭建，降低开发难度，使程序逻辑更加清晰（以 ESP32 彩色 86 盒为例）。



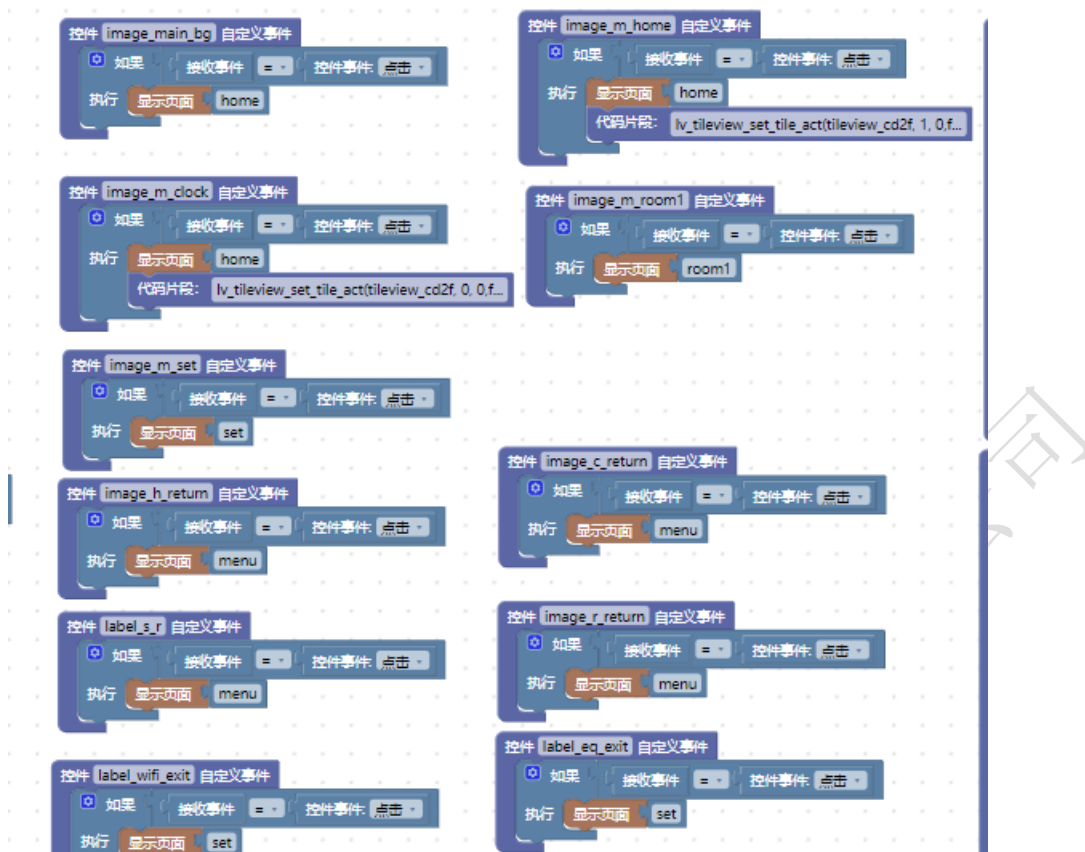
事件闪烁逻辑，通过自定义标志变量，使用定时器定时判断标志实现闪烁。



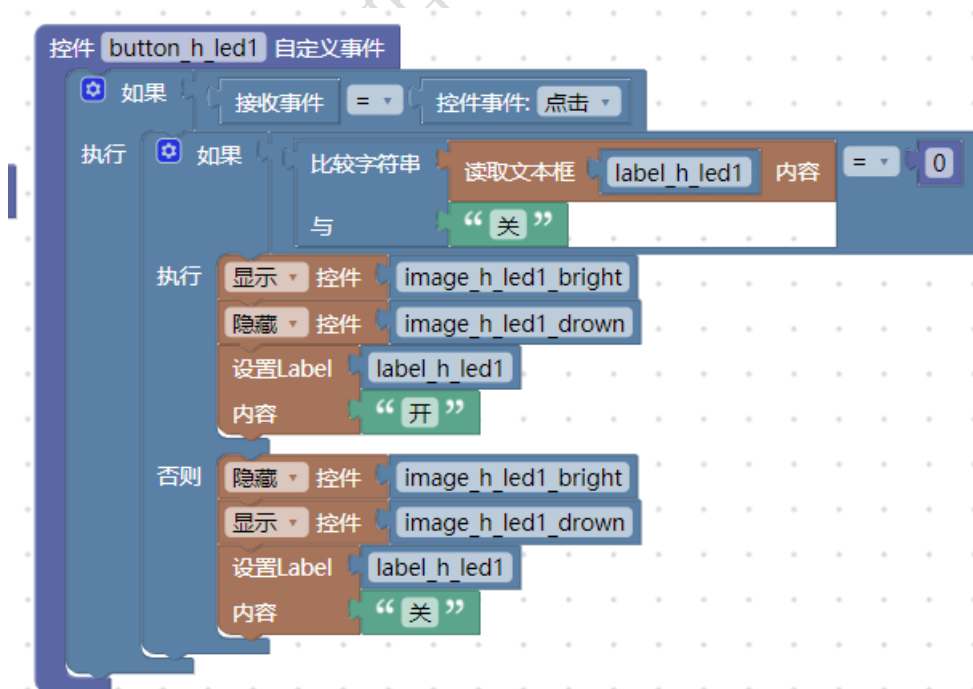
图片旋转逻辑，使用定时器，定时旋转图片角度实现。



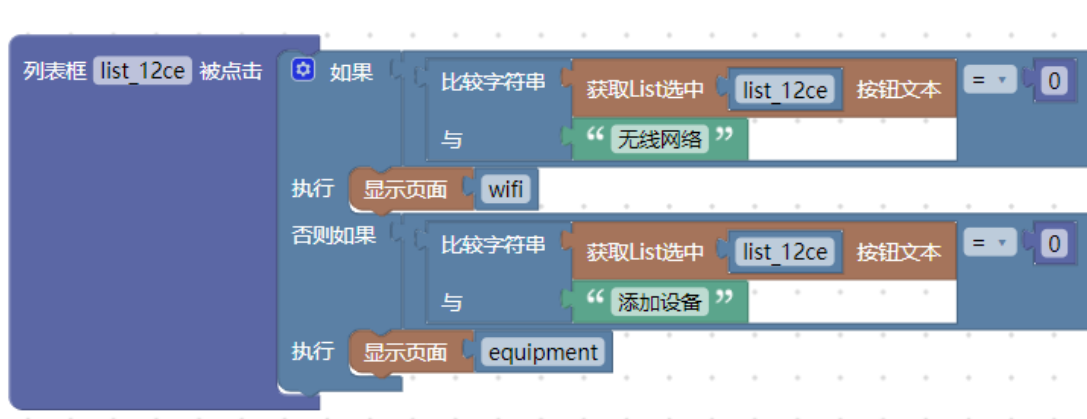
场景切换逻辑，添加 UI 事件，判断点击时跳转到相应的页面。



LED 状态控制逻辑，通过 UI 事件，判断点击时，读取文本内容比较控制 LED 状态。

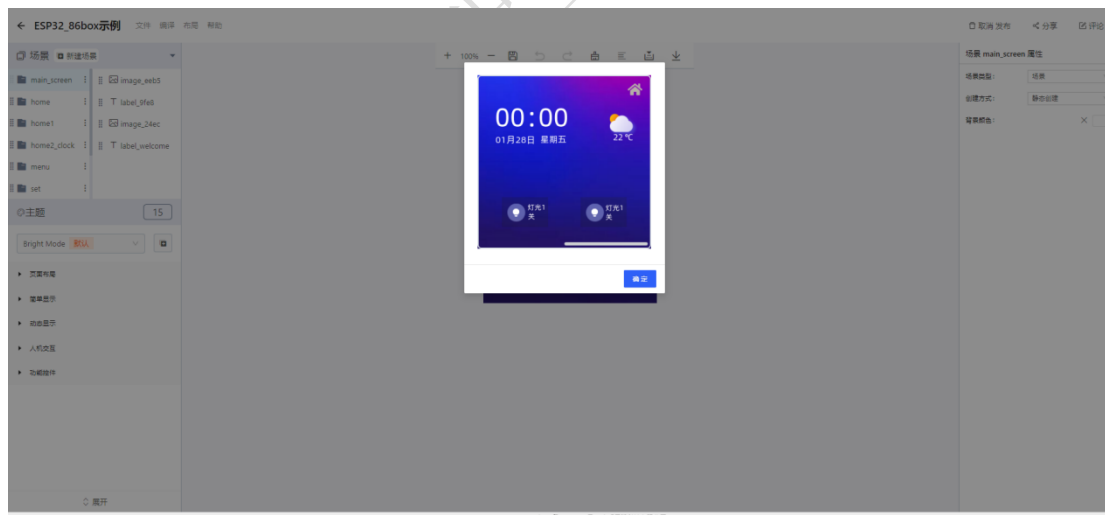


列表点击逻辑，通过 UI 事件，判断读取的按钮文本进行比较实现页面跳转。



四、在线预览

在线模拟预览，执行相同的代码，预览效果与实机一致，方便开发者进行调试。

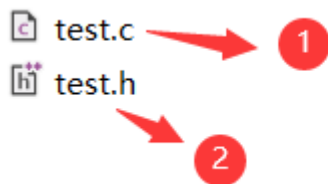


相应的 UI 设计和积木搭建完成后点击编译（在线预览），编译成功后点击预览即可。

五、上传自定义代码



1、编写自定义代码



2021/5/28 14:50

2021/5/28 14:36

① test.c 内容

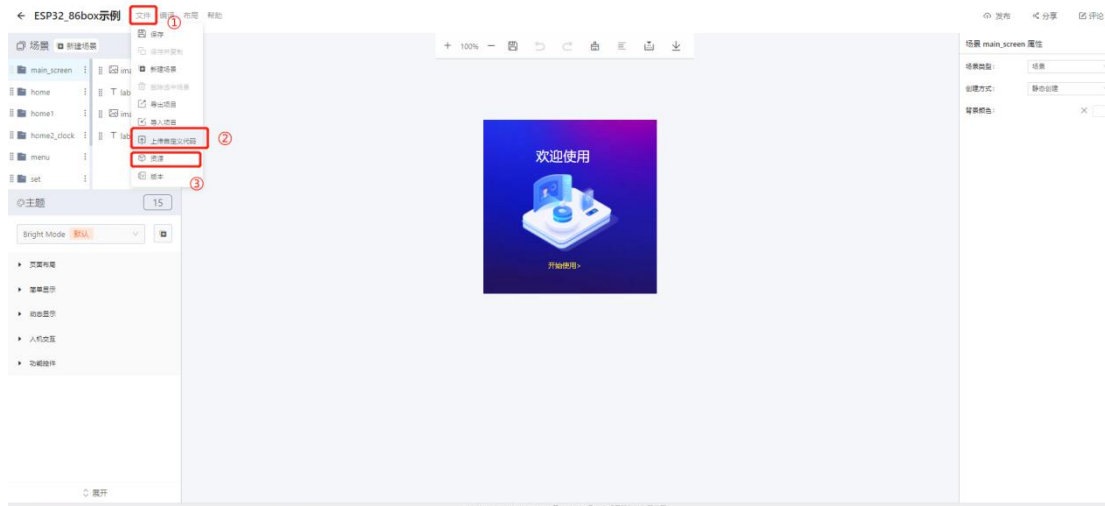
```
1  #include <stdio.h>
2  #include "test.h"
3
4  void funtion(void)
5  {
6      int i=1;
7      printf("i=%d\n",i++);
8  }
9
10
11
```

② test.h 内容

```
1  #ifndef TEST_H
2  #define TEST_H
3
4  void funtion(void);
5
6  #endif
7
8
9
```

2、上传自定义代码到工程

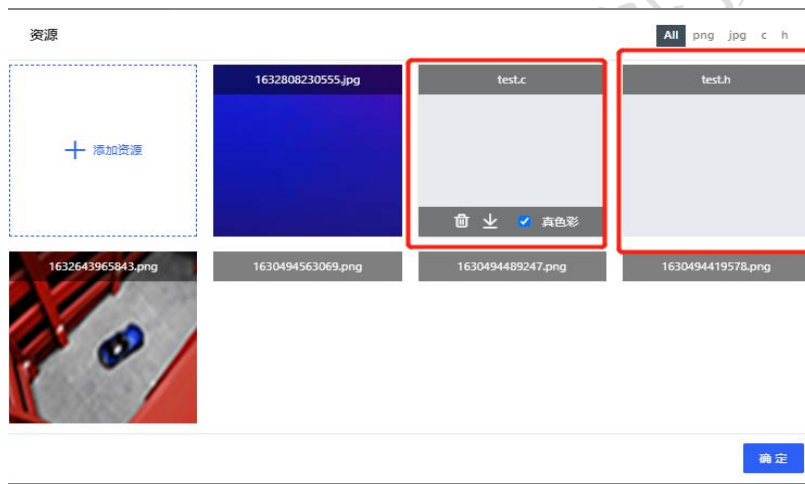
通过自定义代码，可以在本地先将对应的代码逻辑编写好，然后通过自定义上传到平台。



① 点击文件展开文件列表。

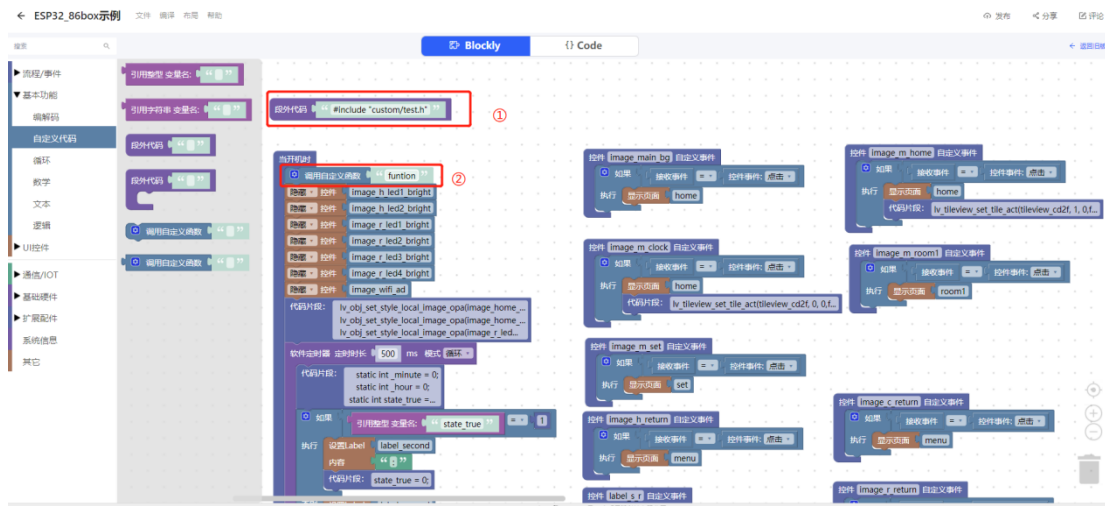
② 点击上传自定义代码，选择编写好的自定义代码。

③ 点击资源，可查看上传的自定义代码（如下图所示）。



点击删除可删除上传的自定义代码。

3、头文件引用和函数调用



① 点击基本功能->自定义代码段->段外代码积木添加 test.h 头文件。

② 点击基本功能->自定义代码段->调用自定义函数积木调用 funtion 函数，可点击积木蓝色设置按钮添加参数。