

### 1，方框选择实现

通过一个绘制函数和覆盖函数实现，按下按键后，先执行覆盖函数，然后执行绘制函数，根据按键对于 X 或者 Y 坐标进行加减，加减的多少根据整体图标设计长和宽以及分辨率大小确定

### 2，json 解析的优化

由于合并了多个程序，内存往往存在不够的情况，特别是 json 解析这里，可以利用 esp8266 的片外 ram 区处理数据。片外 ram 适合大数据量但是非实时性的需求处理。

#### （1）重写系统调用函数

```
struct SpiRamAllocator {  
    void* allocate(size_t size) {  
        return heap_caps_malloc(size, MALLOC_CAP_SPIRAM);  
    }  
    void deallocate(void* pointer) {  
        heap_caps_free(pointer);  
    }  
};  
  
using SpiRamJsonDocument = BasicJsonDocument<SpiRamAllocator>;
```

#### （2）更改调用方式

比如以前的 `DynamicJsonDocument docbuffer(4096);`  
改成 `SpiRamJsonDocument docbuffer(1024*1024*2);`

这样会将以前的几 k 数据的处理变成几 M

### 3，电子书预读取简单说明

翻页时自动加载最新页，然后趁着用户看书的时间后台处理下一页，这样大文本 txt 打开时就不会因为处理耗时过长影响阅读体验了。