

LUA脚本

物联型串口屏通过 LUA 脚本配合工程可以完成大部分的内部逻辑处理，可以做到让MCU 只参与数据处理，不参与屏的逻辑处理。

文本显示函数

1. 函数 set_text (screen, control, str)：指定某个画面中的文本控件或者二维码控件显示指定内容。
2. 函数 on_screen_change(screen)：当画面切换至目标画面 ID 时，执行此回调函数。

回调函数

1. 函数 on_control_notify(screen,control,value)：此函数可以在用户触摸控件后被回调。
2. 函数 on_init()：系统初始化时，执行此回调函数。
3. 函数 on_systick()：定时回调函数，系统每隔 1 秒钟自动调用。
4. 函数 on_timer(timer_id)：定时器超时，执行此回调函数。参数： timer_id --定时器 ID。
5. 函数 on_screen_change(screen)：当画面切换至目标画面 ID 时，执行此回调函数。
6. 函数 on_draw(screen)：当调用 redraw () 或屏幕刷新时会触发此函数，在目标画面中进行绘图。
7. 函数 on_usb_inserted (dir)：当插入 U 盘后会触发调用此函数。参数： dir 表示系统分派给 U 盘的路径。
8. 函数 on_sd_inserted(dir)：当插入 SD 卡后会触发调用此函数。参数： dir 表示系统分派给 SD 卡的路径。
9. 函数 on_usb_removed()：当拔出 U 盘后会触发调用此函数。
10. 函数 on_sd_removed()：当拔出 SD 卡后会触发调用此函。

系统参数函数

1. 函数 on_control_notify(screen,control,value)：此函数可以在用户触摸修改控件后被回调。
2. 函数 uart_set_baudrate(baudrate)，baudrate 表示设置的波特率值。
3. 函数 set_backlight(level)，level 表示设置的背光值。
4. 函数 beep(time)，time 表示蜂鸣器响的时间长度，单位 ms。
5. 函数 set_date_time(year,month,day,hour,minute,second)：此函数可以在用户触摸修改控件后被回调。（注：虚拟屏不支持修改RTC控件，因为是电脑PC端的时间）

定时器函数

1. 函数 on_timer(timer_id)：此函数可以在定时器超时后被回调。timer_id 表示超时定时器 ID。
2. 函数 start_timer(timer_id, timeout, countdown, repeat)：此函数可以启动定时器。
3. 函数 stop_timer(timer_id)：此函数可以关闭定时器。
4. 函数 set_value(screen,control,value)：此函数可以设置控件数值。

控件读写操作函数

1. 函数 get_value(screen,control)：获取控件值，返回的是数值。
2. 函数 get_text(screen,control)：获取控件的字符串内容，文本控件，二维码控件。

3. 函数 `set_value(screen,control,value)`：设置控件数值。
4. 函数 `set_text(screen,control,str)`：设置显示内容，如文本控件、二维码控件。
5. 函数 `set_visiable(screen,control,visiable)`：设置控件隐藏或显示。
6. 函数 `set_enable(screen,control,enable)`：设置触摸使能。
7. 函数 `set_fore_color(screen,control,color)`：设置控件前景色。
8. 函数 `set_back_color(screen,control,color)`：设置控件背景色。

绘图函数

1. 函数 `redraw()`：申请调用回调函数 `on_draw`。
2. 函数 `on_draw(screen)`：当调用 `redraw()` 或屏幕刷新时会触发此函数，在目标画面中进行绘图。
3. 函数 `load_surface (filepath)`：加载图片。
4. 函数 `draw_line(x1,y1,x2,y2,level)`：画直线。

背光亮度函数

1. 函数 `get_backlight()`：此函数获取背光亮度值。
2. 函数 `set_backlight(level)`，`level` 表示背光值：此函数背光亮度。
3. 函数 `io.open (filename [, mode])`：此函数可以打开文件。
4. 函数 `file:write(text)`：此函数可以对文件写入数据。
5. 函数 `file:read([, mode])`：此函数可以读出文件的数据。
6. 函数 `file: seek(optional whence, optional offset)`：此函数设置和获取当前文件位置。
7. 自定义函数 `writefile(filename,info)`，此函数使用 `file:write(text)` 保存数据。
8. 自定义函数 `readfile(filename)`，此函数使用 `file:read([, mode])` 读取数据。