

JUMA API 说明

1 JUMA API 的简单描述

1.1. 与系统状态有关的

回调部分：

```
on_load( void );  
on_ready(void);
```

用户操作部分：

1.2. 与时间相关的操作

回调部分：

回调函数需要用户编写

用户操作部分：

```
void run_when_idle(function_t func, void* args);  
void run_at_time(function_t func, void* args, uint32_t time);  
void run_after_delay(func, args, delay);  
uint32_t current_time(void);
```

1.3. 与锁有关的操作

回调部分：

```
on_lock_state_changed(uint8_t is_locked);
```

用户操作的部分：

无

1.4. 与 GPIO 有关的操作

回调部分：

```
gpio_on_change(uint8_t new_state);
```

用户操作部分：

```
void gpio_setup(uint8_t pin, uint8_t mode);
```

```
void gpio_write(uint8_t pin, uint8_t state);  
uint8_t gpio_read(uint8_t pin);  
void gpio_watch(uint8_t pin, uint8_t change_direction);  
void gpio_unwatch(uint8_t pin);
```

1.5. 与 ADC 有关的操作

回调部分：

回调的函数需要用户来编写

用户操作的部分：

```
void adc_measure(uint8_t pin, uint8_t bits, function_t on_complete);
```

使用说明：

在使用 ADC 的时候需要通过 `adc_measure` 向系统注册一个用于处理数据采集完成的回调函数、用于处理数据采集后的业务逻辑。

其中用于注册的函数形式如下

```
void on_adc_complete(void* args)  
{  
    adc_result_t *result = (adc_result_t*)args;  
    uint8_t buffer[2];  
  
    buffer[0] = result->value >> 8;  
    buffer[1] = result->value & 0xff;  
  
    ble_device_send(buffer, 2);  
}
```

在本函数中，`result` 就是 ADC 采集到的原始的数据。

1.6. 与 BLE 设备相关的操作

回调部分：

```
ble_device_on_connect(void);  
ble_device_on_disconnect(uint8_t reason);  
ble_device_on_message(uint8_t* data, uint32_t size);
```

用户操作的部分：

```
uint8_t ble_device_is_connected(void);  
void ble_device_set_name(const char* device_name);  
void ble_device_disconnect(void);  
void ble_device_send(uint8_t* data, uint32_t size);
```

1.7. 与外接主机相关的操作

回调部分：

```
void hif_on_message(uint8_t* data, uint32_t size);
```

用户操作的部分：

```
void hif_setup(uint8_t rx_pin, uint8_t tx_pin);
```

```
void hif_send(uint8_t* data, uint32_t size);
```

1.8. 与蜂鸣器相关的操作

回调部分：

无

用户操作的部分：

```
void play_sound(uint8_t pin);
```

1.9. 与 RGB 灯光相关的操作

回调部分：

无

用户操作的部分：

```
void light_setup(uint8_t* pins, uint8_t is_active_high);
```

```
void light_on(void);
```

```
void light_off(void);
```

```
void light_set_color(const uint8_t* rgb_values);
```

1.10. 与 Beacon 广播包相关的部分

回调部分：

无

用户操作的部分：

```
void beacon_setup(uint8_t* uuid, uint16_t major, uint16_t minor, uint8_t rssi);
```

2. JUMA API 使用说明

2.1. void **on_load**(void);

函数调用：由系统在板子上电时调用

函数功能：处理需要在板子上电时的事物（如指定蓝牙模块的名字）

函数特点：必须实现，并且不能为空

参数：无

返回：无

2.2. void **on_ready**(void);

函数调用：由系统在蓝牙模块已经开始发送广播后调用

函数功能：处理在开始广播后的事物

函数特点：

参数：无

返回：无

2.3. void **on_lock_state_changed**(uint8_t is_locked);

函数调用：由系统在锁的状态被改变后调用

函数功能：当锁状态改变后处理其他的一些东西

函数特点：

参数：当前锁的状态

返回：无

2.4. void **gpio_on_change**(uint8_t new_state);

函数调用：由系统在外中断 GPIO 的状态被改变后调用

函数功能：当接收到外部中断事件发生后的处理

函数特点：

参数：当前引脚的状态

返回：无

2.5. void **ble_device_on_connect**(void);

函数调用：当有 BLE 设备连接本设备后被调用

函数功能：处理其他的 BLE 设备连接后的任务

函数特点：

参数：无

返回：无

2.6. void **ble_device_on_disconnect** (uint8_t reason);

函数调用：当原来连接的 BLE 设备被断开后被调用

函数功能：处理原来连接的 BLE 设备断开后的任务

函数特点：

参数：无

返回：无

2.7. void **ble_device_on_message**(uint8_t* data, uint32_t size);

函数调用：当与之相连的 BLE 设备向本设备发送消息时被调用

函数功能：处理其他 BLE 设备发过来的消息

函数特点：

参数：

Data 表示收到的消息

Size 表示收到的消息的大小

返回：无

2.8. void **hif_on_message** (uint8_t* data, uint32_t size);

函数调用：当接收到来自串口的消息后被调用

函数功能：处理来自串口的消息

函数特点：

参数：

Data 收到的数据

Size 收到数据的大小

返回：无

2.9. void **run_when_idle**(function_t func, void* args);

函数调用：由用户使用

函数功能：向系统注册一个在系统空闲的时候运行的函数

函数特点：要执行的函数对时间要求不高

参数：

func 要执行的函数的位置（函数指针）

args 执行函数时传入函数的参数（只能是一个指针）

返回：无

2.10. void **run_at_time**(function_t func, void* args, uint32_t time);

函数调用：由用户使用

函数功能：向系统注册一个在指定的时间点上运行的函数

函数特点：要执行的函数对时间点有要求

参数：

func 要执行的函数的位置（函数指针）

args 执行函数的时候传入函数的参数（只能是一个指针）

time 函数执行的时间点

返回：无

2.11. void **run_after_delay**(func, args, delay);(内部使用宏实现)

函数调用：由用户调用

函数功能：向系统注册一个在指定时间后运行的函数

函数特点：要执行的操作对时间点有要求

事实上这是一个由 **run_at_time** 和 **current_time** 构成的宏

参数：

func 要执行的函数的位置（函数指针）

args 执行函数的时候传入的参数（只能是一个指针）

time 等待的时间长度

返回：无

2.12. uint32_t **current_time**(void);

函数调用：由用户调用

函数功能：获取当前时间计数器值

函数特点:

参数: 无

返回: 无

2.13. void **gpio_setup** (uint8_t pin, uint8_t mode);

函数调用: 由用户调用

函数功能: 用于对一个 GPIO 引脚的功能进行设定 (输出 or 输入)

函数特点:

参数:

pin	pin 值
mode	指定该引脚用作的模式(在 juma_sdk_types.h 中有定义如下) GPIO_OUTPUT, GPIO_INPUT_NOPULL, GPIO_INPUT_PULLUP, GPIO_INPUT_PULLEDOWN,

返回: 无

2.14. void **gpio_write** (uint8_t pin, uint8_t state);

函数调用: 由用户调用

函数功能: 设置 gpio 的状态 (非 0 高电平或者 0 低电平)

函数特点:

参数:

pin	pin 值
state	状态 (非 0 or 0)

返回: 无

2.15. uint8_t **gpio_read** (uint8_t pin);

函数调用: 由用户调用

函数功能: 获取指定 pin 脚的状态

函数特点:

参数:

pin	pin 值
-----	-------

返回: 指定的 pin 的状态

2.16. void **gpio_watch** (uint8_t pin, uint8_t change_direction);

函数调用：由用户调用

函数功能：向系统注册一个 GPIO 事件，当满足指定条件的引脚状态变化满足时，on_gpio_pin_state_changed(uint8_t new_state);将会被系统自动调用。

函数特点：用户注册一个 GPIO 的系统事件

参数：

pin	引脚值
chang_direction	变化方式(在 juma_sdk_types.h 中有定义如下) GPIO_RISING , GPIO_FALLING,

返回：无

2.17. void **gpio_unwatch** (void);

函数调用：由用户调用

函数功能：解除已经注册的 GPIO 事件

函数特点：

参数：无

返回：无

2.18. void **adc_measure**(uint8_t pin, uint8_t bits, function_t on_complete);

函数调用：由用户调用

函数功能：获取指定的 PIN 脚上的模拟信号，并在获取完成后自动调用用户指定的函数 on_complete 来处理获取到的模拟量

函数特点：用户需要指定 pin 和相应的处理函数，当模拟量获取完毕后系统自动调用处理函数

参数：

pin	pin 值(可用的取值:1/2/3/4/5/26/27)
bits	指定采集的精度（有 8:0-255/ 9:0-511/10:0-1023 可选）
on_complete	模拟信号采集完毕后系统要执行的函数（函数指针） 其格式(在 juma_sdk_types.h 中有定义如下) void (*function_t)(void* args);

返回：无

2.19. uint8_t **ble_device_is_connected**(void);

函数调用：由用户调用

函数功能：判断当前是否有 BLE 设备与自身连接

函数特点：

参数：无

返回：是否有 BLE 设备与自身连接

2.20. void **ble_device_set_name**(const char* device_name);

函数调用：由用户调用

函数功能：设置当前的 BLE 模块的名字

函数特点：

参数：

Device_name 要设置的名字的字符串

返回：无

2.21. void **ble_device_disconnect**(void);

函数调用：由用户调用

函数功能：断开与当前的 BLE 模块的连接

函数特点：

参数：无

返回：无

2.22. void **ble_device_send**(uint8_t* data, uint32_t size);

函数调用：由用户调用

函数功能：向与之相连的 BLE 设备发送消息

函数特点：

参数：

Data 要发送的消息的起始地址

Size 要发送的消息的长度

返回：无

2.23. void **hif_setup**(uint8_t rx_pin, uint8_t tx_pin);;

函数调用：由用户调用

函数功能：设置串口的基本参数，串口用作与外挂主机通信

函数特点：

参数：

rx_pin 指定串口的 RX 脚

tx_pin 指定串口的 TX 脚

返回：无

2.24. void **hif_send**(uint8_t* data, uint32_t size);

函数调用：由用户调用

函数功能：通过串口发送数据,串口用作与外挂主机通信

函数特点：

参数：

data 要发送的数据

size 数据大小

返回：无

2.25. void **play_sound**(uint8_t pin);;

函数调用：由用户调用

函数功能：让蜂鸣器开始响一下

函数特点：

参数：

pin 与蜂鸣器连接的引脚

返回：无

2.26. void **light_setup** (uint8_t* pins, uint8_t is_active_high);

函数调用：由用户调用

函数功能：设置 RGB 的 light

函数特点：

参数：

pins 引脚数组(在 juma_sdk_types.h 中有定义如下)
uint8_t pins[4]; // pins for RGBW

is_active_high 0: 低电平驱动 非 0: 高电平驱动
返回: 无

2.27. void **light_on**(void);

函数调用: 由用户调用
函数功能: 打开 RGB 灯
函数特点:

参数: 无
返回: 无

2.28. void **light_off**(void);

函数调用: 由用户调用
函数功能: 关闭 RGB 灯
函数特点:

参数: 无
返回: 无

2.29. void **light_set_color** (const uint8_t* rgb_values);

函数调用: 由用户调用
函数功能: 设置 RGB 的颜色
函数特点:

参数:
 rgb_values rgb 的值
返回: 无

2.30. void **beacon_setup**(uint8_t* uuid, uint16_t major, uint16_t minor, uint8_t rssi);

函数调用: 由用户调用
函数功能: 设置 beacon 的参数
函数特点:

参数:

uuid	beacon 的 UUID
major	major
minor	minor
rss	在距离发射源 1m 处的测量值（用于辅助测距）
返回：	无