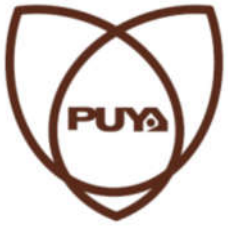
PY32T020xx-触摸库

用户指南

PUYA CONFIDENTIAL



Puya Semiconductor (Shanghai) Co., Ltd

目录

PUYA CONFIDENTIAL

1. 工程目录简介

1.1 触摸库工程目录简介

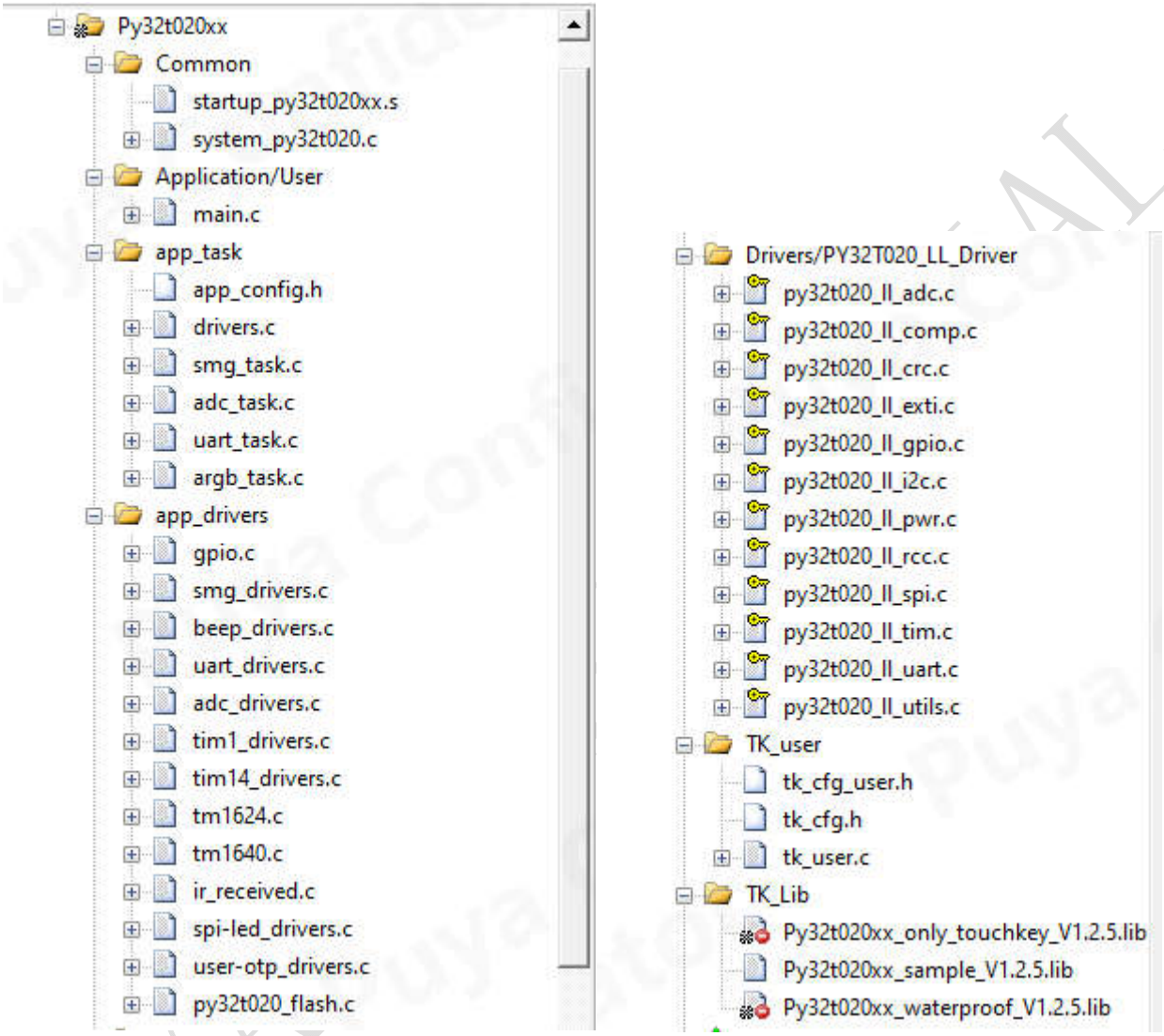


图 1-1

Common	PY32T020XX 启动文件
Application/User	main 文件，程序入口
app_task	示例代码应用层
app_drivers	基于 LL 库再次封装的代码底层驱动
Drivers/PY32T020_LL_Driver	PY32T020_LL 库驱动
TK_user	触摸用户初始化文件以及回调函数
TK_Lib	触摸库

1.2 触摸库版本介绍

1. Py32t020xx_only_touchkey_Vx.x.x.lib 仅支持触摸按键，专为小容量芯片定制，占用更少的 RAM 以及 ROM，以下简称 mini 库；
2. Py32t020xx_sample_Vx.x.x.lib 支持触摸按键以及触摸滑条（触摸圆环），适用于大多数的应用，以下简称 sample 库；
3. Py32t020xx_waterproof_Vx.x.x.lib 支持按键，圆环，滑条，高灵敏度触摸，防水功能；以下简称标准库；

1.3 触摸库互相切换方法

1) 根据需要替换 lib 文件；
2) 在 Options 中修改宏定义，如图 1-2 所示，当使用 mini 库时 LIB_TYPE = 0，使用 sample 库时 LIB_TYPE = 1,使用标准库时，LIB_TYPE = 2;

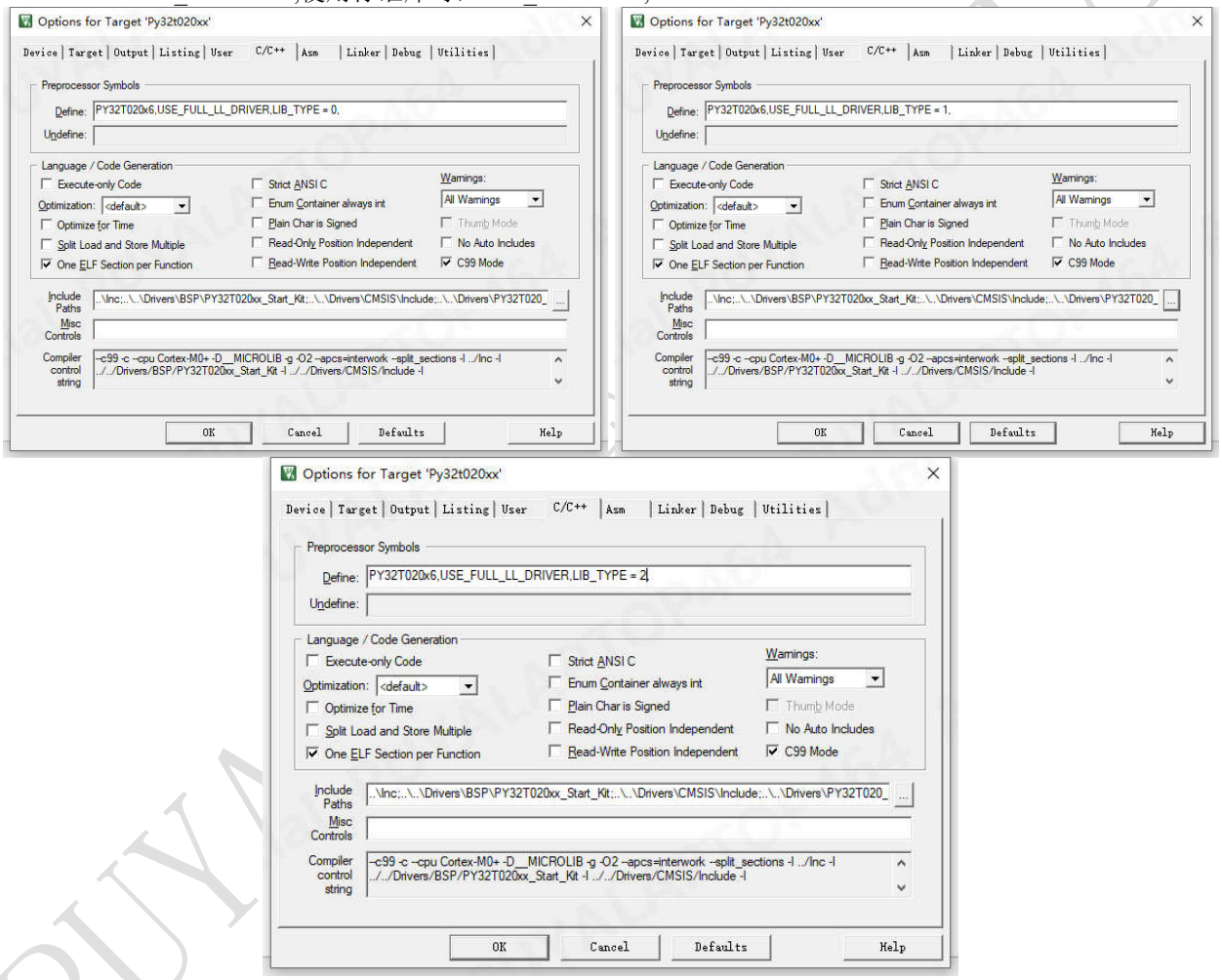


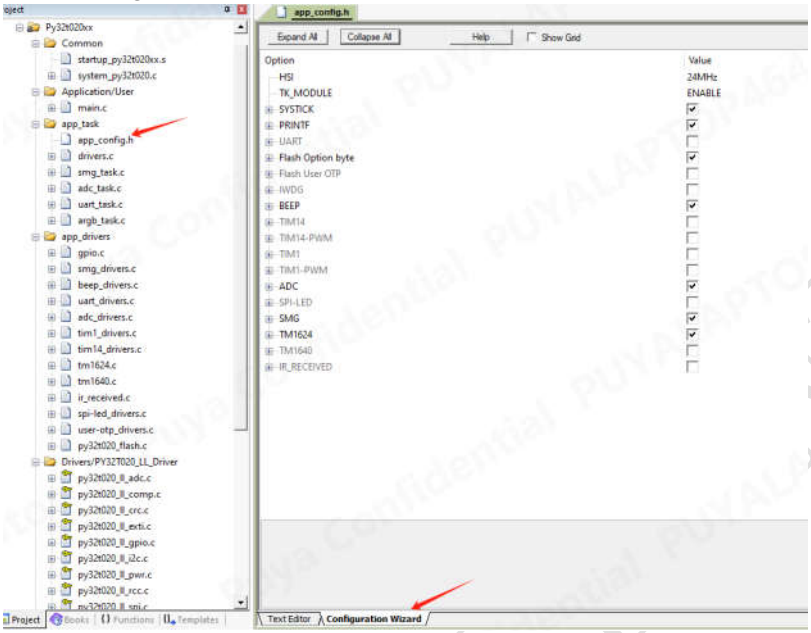
图 1-2

2. 功能配置引导

2.1 外设库使用方法

1、 打开 app_config.h

2、 点击 Configuration Wizard,会出现如图界面



3、 可以通过该窗口进行底层驱动的配置

HSI	配置系统时钟频率，默认 24M
TK_MODULE	触摸库使能开关
SYSTICK	配置系统定时器定时时间，触摸库以及外设需要 1ms 定时基准，所以设置的时间需要能被 1ms 整除
PRINTF	用于开启打印调试信息，使用上位机调试需开启触摸库
UART	用于开启 UART 模块，3 路串口可以使用，目前演示功能为通过 UART 回发收到的数据
Flash Option byte	系统选项字节配置，配置 PF2 功能，低电压复位，看门狗状态
Flash User OTP	使用 FLASH 保存用户数据，共 124 个字节可用
IWDG	用于开启看门狗以及设置看门狗超时时间
BEEP	用于开启蜂鸣器以及设置蜂鸣器 GPIO、电平模式
TIM14	使用 TIM14 进行定时功能，可配置定时时间，中断函数为 TIM14_PeriodElapsedCallback
TIM14-PWM	使用 TIM14 进行 PWM 功能，可配置 PWM 频率，分辨率，输出管脚
TIM1	使用 TIM1 进行定时功能，可配置定时时间，中断函数为 TIM1_PeriodElapsedCallback
TIM1-PWM	使用 TIM1 进行 PWM 功能，可配置 PWM 频率，分辨率，输出管脚
ADC	用于开启 ADC 检测，默认为 12 位分辨率，参考电压为 VCC，可更改采样通道以及采样间隔
SPI-LED	使用 SPI 来驱动类 W2812B 灯珠，可修改输出管脚以及驱动灯珠的数量
SMG	数码管软件扫描，可设置 SEG 管脚以及 COM 管脚，驱动共阴极数码管,从 COM0 跟 SEG0 开始填，后面没有选择 NO_PIN，通过改变 smg_data 来设置显示内容

PUYA CONFIDENTIAL

TM1624	用于驱动 TM1624/1628,可配置 CLK-DIN-STB 管脚以及显示模式、显示亮度，调用 TM1624_Display_Update 进行刷新显示
TM1640	用于驱动 TM1640,可配置 CLK-DIN 管脚以及显示亮度，调用 TM1640_Display_Update 进行刷新显示
IR_RECEIVED	用于解码 NEC 格式的红外编码，需要选择对应的定时器以及数据接收管脚

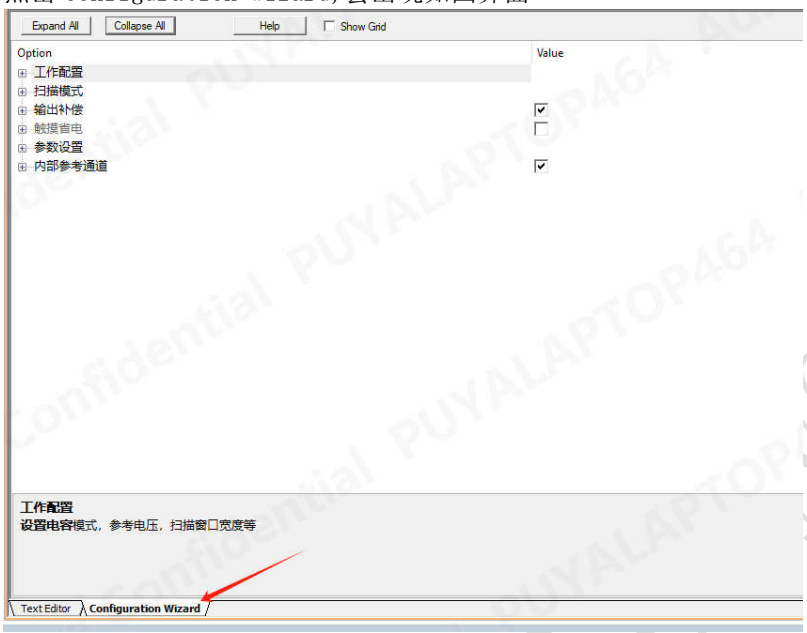
注：具体事项请查看文件内说明

PUYA CONFIDENTIAL

2.2 触摸库使用方法

1、 打开 tk_cfg.h

2、 点击 Configuration Wizard,会出现如图界面



3、 可以通过该窗口进行触摸库的配置

工作配置	CMOD 电容	可选择内部或者外部，正常应用使用内部即可，高低温变化明显的可以使用外部模式
	参考电压	设置触摸充电的参考电压，四挡可选
	扫描窗口	设置采样窗口时间，设置越大，扫描时间越长
	自适应电流	用于开启电流自适应以及灵敏度均衡功能
扫描模式	分频系数	用于设置 SW 频率
	跳频模式	用于开启跳频功能，建议开启，增加抗干扰能力
输出补偿	补偿输出电压	设置补偿输出电压大小
	补偿输出类型	设置补偿的类型
触摸省电	无操作进入休眠时间	设置进入休眠的时间，单位 5ms
	扫描触摸的间隔时间	设置触摸唤醒的扫描间隔，单位 ms
	触摸省电模式下门限值	设置唤醒门限值的大小，当扫描的数据变化超过该值时唤醒触摸
	休眠模式窗口值	设置采样窗口时间，设置越大，扫描时间越长，功耗越高
	参考电压	设置触摸充电的参考电压，四挡可选，电压越高，功耗越大
参数设置	NOISE 门限值	用于设置噪音的大小
	触摸数据滤波次数	范围：1~30，此数值越大，数据越平滑，但是按键响应速度越慢
	限制按键最长输出时间	设置按键最长输出时间，当按键长按超过该时间则自动释放按键
	触摸按键按下消抖时间	按键按下消抖时间
	触摸按键释放消抖时间	按键释放消抖时间
	WATER_AREA 更新基线时间	保持默认即可
	NOISE_AREA 更新基线时间	保持默认即可

PUYA CONFIDENTIAL

	BOTTON_AREA 更新基线时间	保持默认即可
	单键触发	开启后只能同时触发一个按键
	设置多按键数量	当多按键同时触发时强制更新 BASELINE,设置为 0 即功能关闭
内部参考通道	门限值	内部参考通道门限值

PUYA CONFIDENTIAL

3. 外设库函数接口说明

```
**      函数名  void GPIO_Init(uint8_t gpio,uint32_t Init)
**      描述            GPIO 初始化
**      传入    :      gpio：IPA0~PA15,PB0~PB3,PF0~PF5
                       Init：初始化的参数
        将 GPIO 设置为模拟模式          ->      GPIO_Init(PA0,ANALOG)
        将 GPIO 设置为输入上拉模式              ->      GPIO_Init(PA0,INPUT|PULL_UP)
        将 GPIO 设置为输入上拉下降沿中断模式 ->
                                                GPIO_Init(PA0,INPUT|PULL_UP|EXTI_TRIGGER_FALLING)
        将 GPIO 设置为推挽输出模式              ->      GPIO_Init(PA0,OUTPUT|PUSHPULL)
        将 GPIO 设置为开漏输出模式              ->      GPIO_Init(PA0,OUTPUT|OPENDRAIN)
        将 GPIO 设置为 TIM1_CH3 输出            ->      GPIO_Init(PA0,ALTERNATE | GPIO_TIM1_AF2)
                       (这里选择 GPIO_TIM1_AF2 或者 GPIO_TIM1_AF5 需要查看规格书的复用功能映射)
        将 GPIO 设置为 UART 功能                ->      GPIO_Init(PA0,ALTERNATE | GPIO_UART2)
**      返回    : 无

**      函数名  void GPIO_SetBit(uint8_t gpio)
**      描述            GPIO 输出高电平
**      传入    :      gpio：PA0~PA15,PB0~PB3,PF0~PF5
**      返回    : 无

**      函数名  void GPIO_ClearBit(uint8_t gpio)
**      描述            GPIO 输出低电平
**      传入    :      gpio：PA0~PA15,PB0~PB3,PF0~PF5
**      返回    : 无

**      函数名  void GPIO_ToggleBit(uint8_t gpio)
**      描述    GPIO 输出翻转
**      传入    :      gpio：PA0~PA15,PB0~PB3,PF0~PF5
**      返回    : 无

**      函数名  uint8_t GPIO_ReadBit(uint8_t gpio)
**      描述    读取 GPIO 状态
**      传入    :      gpio：PA0~PA15,PB0~PB3,PF0~PF5
**      返回    : gpio 状态 0 或 1

**      函数名  void EXTI0_15_IRQHandlerCallback(uint32_t PR)
**      描述    外部中断 0 到 15 的回调函数
**      传入    : PR:中断标志，BIT0 为外部中断 0，BIT1 为外部中断 1，以此类推
**      返回    : 无

**      函数名  void ADC_GPIO_Init(void)
**      描述    : 客户补全，将单独采样的 GPIO 设置为模拟输入模式，如 GPIO_Init(PA0, ANALOG);
**      传入    : 无
**      返回    : 无

**      函数名  void ADC_Init(void)
**      描述    :      ADC 初始化（采样通道，采样间隔通过 app_config.h 进行配置）
**      传入    :      无
**      返回    :      无
```

PUYA CONFIDENTIAL

```
**      函数名  uint16_t APP_ADCConvert(uint16_t channel, uint32_t VrefBuf)
**      描述    ：ADC 单次采集，主要用于参考电压不是 VCC 的情况
**      传入    ：      channel：通道号
                            ADC_CHANNEL_0 ~ ADC_CHANNEL_12
                    VrefBuf：设置的参考电压
                            ADC_VREFBUF_VCCA
                            ADC_VREFBUF_0P6V
                            ADC_VREFBUF_1P5V
                            ADC_VREFBUF_2P048V
                            ADC_VREFBUF_2P5V
**      返回    ：采样到的 12 位的 ADC 数据

**      函数名  void ADC_Loop(void)
**      描述    ADC 状态机，当 adc_state 为 2 时表示序列转换完成（ADC_ENABLE_CHS 内包含的所有通
道），可以直接从 ADCxConvertedData[x]中取数据，x 为通道号
**      传入    ：无
**      返回    ：无

**      函数名  void UART1_Init(uint32_t BaudRate)
**      描述    ：UART1 初始化函数（发送接收管脚通过 app_config.h 进行配置）
**      传入    ：BaudRate 波特率
**      返回    ：      无

**      函数名  uint8_t UART1_QueueRead(uint8_t *data)
**      描述    ：UART1 读取数据
**      传入    ：data：接收数据指针
**      返回    ：      0 数据为空
                        1 数据有效

**      函数名  uint8_t UART1_QueueSend(uint8_t data)
**      描述    ：UART1 发送数据
**      传入    ：data：发送的数据
**      返回    ：      0 发送失败
                        1 发送成功

**      函数名  void UART2_Init(uint32_t BaudRate)
**      描述    ：UART2 初始化函数（发送接收管脚通过 app_config.h 进行配置）
**      传入    ：BaudRate 波特率
**      返回    ：      无

**      函数名  uint8_t UART2_QueueRead(uint8_t *data)
**      描述    ：UART2 读取数据
**      传入    ：data：接收数据指针
**      返回    ：      0 数据为空
                        1 数据有效

**      函数名  uint8_t UART2_QueueSend(uint8_t data)
**      描述    ：UART2 发送数据
**      传入    ：data：发送的数据
**      返回    ：      0 发送失败
                        1 发送成功

**      函数名  void UART3_Init(uint32_t BaudRate)
**      描述    ：UART3 初始化函数（发送接收管脚通过 app_config.h 进行配置）
**      传入    ：BaudRate 波特率
```

PUYA CONFIDENTIAL

```
**      返回    :       无

**      函数名  uint8_t UART3_QueueRead(uint8_t *data)
**      描述    : UART3 读取数据
**      传入    : data：接收数据指针
**      返回    :       0 数据为空
                        1 数据有效

**      函数名  uint8_t UART3_QueueSend(uint8_t data)
**      描述    : UART3 发送数据
**      传入    : data：发送的数据
**      返回    :       0 发送失败
                        1 发送成功

**      函数名  void UART_Loop(void)
**      描述    UART 任务处理，轮询 UART 接收队列，当接收空闲超过 5ms 或者接收数据超过 64 字节表示
接收结束，回发收到的数据
**      传入    : 无
**      返回    : 无

**      函数名  void TIM1_Init(void)
**      描述    定时器 1 初始化（定时时间通过 app_config.h 进行配置）
**      传入    : 无
**      返回    : 无

**      函数名  void TIM1_PeriodElapsedCallback(void)
**      描述    定时器 1 中断回调函数
**      传入    : 无
**      返回    : 无

**      函数名  void TIM1_PWM_Init(void)
**      描述    定时器 1 用做 PWM 初始化
                        （分辨率，频率，输出管脚通过 app_config.h 进行配置）
**      传入    : 无
**      返回    : 无

**      函数名  void TIM1_PWM_Pulse(uint32_t CHx,uint16_t percent)
**      描述    定时器 1-PWM 占空比设置
**      传入    :       CHx 通道号      LL_TIM_CHANNEL_CH1~LL_TIM_CHANNEL_CH4，
                        percent         输出百分比，最大值为 TIM1_RESOLUTION
**      返回    : 无

**      函数名  void TIM14_Init(void)
**      描述    定时器 14 初始化（定时时间通过 app_config.h 进行配置）
**      传入    : 无
**      返回    : 无

**      函数名  void TIM14_PeriodElapsedCallback(void)
**      描述    定时器 14 中断回调函数
**      传入    : 无
**      返回    : 无

**      函数名  void TIM14_PWM_Init(void)
**      描述    定时器 1 用做 PWM 初始化
```

PUYA CONFIDENTIAL

（分辨率，频率，输出管脚通过 app_config.h 进行配置）
** 传入 ：无
** 返回 ：无

** 函数名 void TIM14_PWM_Pulse(uint32_t CHx,uint16_t percent)
** 描述 定时器 14-PWM 占空比设置
** 传入 ： CHx 通道号 LL_TIM_CHANNEL_CH1
percent 输出百分比，最大值为 TIM14_RESOLUTION
** 返回 ：无

** 函数名 uint8_t User_Cache_Read(uint8_t offset,uint8_t *data,uint8_t len)
** 描述 从缓存中读取数据，总共 124 个字节可使用
** 传入 ： offset，偏移量 data:读取的数据指针，len:读取的数据长度
** 返回 ： 0：数据无效或长度超出
1：数据读取成功

** 函数名 uint8_t User_Cache_Write(uint8_t offset,uint8_t *data,uint8_t len)
** 描述 写入数据到缓存中，总共 124 个字节可使用
** 传入 ： offset，偏移量 data:写入的数据指针，len:读取的数据长度
** 返回 ： 0：长度超出
1：数据写入成功

** 函数名 void User_Flash_Write(void)
** 描述 将缓存内数据写入 FLASH 中
** 传入 ： 无
** 返回 ： 0：保存失败
1：保存成功

** 函数名 void SPI_LED_Init(void)
** 描述 SPI 初始化，用于驱动类 W2812 灯珠，
（输出管脚，灯珠数量通过 app_config.h 进行配置）
通过 WS2812_1_CODE 以及 WS2812_0_CODE 可更改时序
** 传入 ： 无
** 返回 ： 无

** 函数名 void SPI_LED_RgbLoad(uint8_t offset, uint8_t red, uint8_t green, uint8_t blue)
** 描述 将 RGB 数据写入缓存中
** 传入 ： offset：灯珠偏移量
red:红灯 PWM 值
green:绿灯 PWM 值
blue:蓝灯 PWM 值
** 返回 ： 无

** 函数名 void SPI_LED_Transmit(void)
** 描述 将缓存数据通过 SPI 发送
** 传入 ： 无
** 返回 ： 无

** 函数名 void IR_Received_Init(void)
** 描述 红外扫描解码初始化（信号输入管脚，扫描用的定时器通过 app_config.h 进行配置）
** 传入 ：无
** 返回 ：无

** 函数名 void IR_Received_Scan(void)

PUYA CONFIDENTIAL

```
**      描述    红外解码电平扫描函数，放在定时器中断内
**      传入    ：无
**      返回    ：无

**      函数名  uint8_t IR_Press(Ir_TypeDef *remote)
**      描述    红外解码函数
**      传入    ：remote 数据接收结构体
**      返回    ：      0：无信号输入
                        1：收到正确的红外信号

**      函数名  void SMG_Init(void)
**      描述    数码管 IO 口初始化（SEG,COM 所用的 GPIO 通过 app_config.h 进行配置）
**      传入    ：无
**      返回    ：无

**      函数名  void SMG_Sleep(void)
**      描述    数码管休眠，设置 IO 口状态
**      传入    ：无
**      返回    ：无

**      函数名  void SMG_Wake(void)
**      描述    数码管退出休眠，继续显示
**      传入    ：无
**      返回    ：无

**      函数名  void SMG_Scan(void)
**      描述    数码管扫描，定时器内调用
**      传入    ：无
**      返回    ：无

**      函数名  void TM1624_Init(void)
**      描述    TM1624 初始化（CLK,DIN,STB 管脚，显示模式，显示亮度通过 app_config.h 进行配置）
**      传入    ：无
**      返回    ：无

**      函数名  void TM1624_Display_Update(uint8_t *display,uint8_t num)
**      描述    TM1624 刷新
**      传入    ：display 显示的数据指针 num 显示的数量 Config 命令指令，用于亮度调节以及开关显示
**      返回    ：无

**      函数名  uint8_t TM1624_Read_Key(uint8_t *data)
**      描述    TM1628 读取按键,1624 无该功能
**      传入    ：data 按键指针
**      返回    ：无

**      函数名  void TM1640_Init(void)
**      描述    TM1640 初始化（CLK,DIN 管脚，显示亮度通过 app_config.h 进行配置）
**      传入    ：无
**      返回    ：无

**      函数名  void TM1640_Display_Update(uint8_t *display,uint8_t num,uint8_t Config)
**      描述    TM1640 刷新
**      传入    ：display 显示的数据指针 num 显示的数量 Config 命令指令，用于亮度调节以及开关显示
**      返回    ：无
```

PUYA CONFIDENTIAL

4. 触摸库函数接口说明

```
**      函数名  void TK_Init(void)
**      描述    触摸库初始化
**      传入    ：无
**      返回    ：无

**      函数名  uint32_t TK_GetVersion(void)
**      描述    获取触摸库版本
**      传入    ：无
**      返回    ：触摸库版本

**      函数名  void TK_MainFsm (void)
**      描述    触摸库状态机处理，主函数内调用
**      传入    ：无
**      返回    ：无

**      函数名  void TK_TimerHandler(uint8_t ms)
**      描述    触摸库定时器
**      传入    ：ms，当前时基
**      返回    ：无

**      函数名  void TK_UserParaInit (void)
**      描述    TK 用户参数初始化，该函数无需修改以及调用，保持默认即可
**      传入    ：无
**      返回    ：无

**      函数名  void TK_RegisterCfg (void)
**      描述    TK 寄存器初始化，该函数无需修改以及调用，保持默认即可
**      传入    ：无
**      返回    ：无

**      函数名  void TK_MainFsm (void)
**      描述    触摸库状态机处理，主函数内调用
**      传入    ：无
**      返回    ：无

**      函数名  void EnterStop_Callback (void)
**      描述    触摸进入休眠前的回调函数，用于关闭不必要的外设
**      传入    ：无
**      返回    ：无

**      函数名  void ExitStop_Callback (void)
**      描述    触摸退出休眠后的回调函数，用于恢复之前关闭的外设
**      传入    ：无
**      返回    ：无

**      函数名  uint8_t APP_TouchKeyFlagsMask(void)
**      描述    TK 按键屏蔽，当有按键触发时库函数调用，
 **     传入    ：无
**      返回    ：      0：无异常
                        1：有异常，需要屏蔽按键
```

PUYA CONFIDENTIAL

```
**      函数名  uint8_t APP_TouchShieldFlagsMask(uint8_t chs, uint16_t BaseLineData, uint16_t AcqData)
**      描述    触摸防水处理
**      传入    :      chs:当前触发的通道
                       BaseLineData：  环境值
                       AcqData：       当前值
**      返回    :      0：无异常
                       1：检测到有水

**      函数名  uint8_t APP_TouchWaterFlagsMask(uint8_t chs, int16_t Differ, int16_t DifferSigle)
**      描述    触摸防水处理
**      传入    :      chs:当前触发的通道
                       Differ：正常采集值
                       DifferSigle：特殊采集值
**      返回    :      0：正常触摸
                       1：有水流事件
                       2：有水时按键触发
```

PUYA CONFIDENTIAL

5. 触摸基础配置说明(tk_cfg_user.h)

此文件主要用于触摸通道及对应通道灵敏度的设置，由用户自行设置，也可由上位机调试工具（PY Touch – Link.exe）导出。

5.1 触摸按键配置

```
#define CH_KEY0          TK_CH0
#define CH_KEY1          TK_CH1
#define CH_KEY2          TK_CH12
#define CH_KEY3          TK_CH6
#define CH_KEY4          TK_CH7
#define CH_KEY5          TK_CH11
#define CH_KEY6          TK_CH8
#define CH_KEY7          TK_CH9
#define CH_KEY8          TK_CH10
#define CH_KEY9          TK_CH_NONE
#define CH_KEY10         TK_CH_NONE
#define CH_KEY11         TK_CH_NONE
#define CH_KEY12         TK_CH_NONE
#define CH_KEY13         TK_CH_NONE
#define CH_KEY14         TK_CH_NONE
#define CH_KEY15         TK_CH_NONE
#define CH_KEY16         TK_CH_NONE
#define CH_KEY17         TK_CH_NONE
#define CH_KEY18         TK_CH_NONE
#define CH_KEY19         TK_CH_NONE
#define CH_KEY20         TK_CH_NONE
#define CH_KEY21         TK_CH_NONE
#define CH_KEY22         TK_CH_NONE
#define CH_KEY23         TK_CH_NONE
#define CH_KEY24         TK_CH_NONE
#define CH_KEY25         TK_CH_NONE
```

```
#define KEY0_THD         50
#define KEY1_THD         50
#define KEY2_THD         50
#define KEY3_THD         50
#define KEY4_THD         50
#define KEY5_THD         50
#define KEY6_THD         50
#define KEY7_THD         50
#define KEY8_THD         50
#define KEY9_THD         40
#define KEY10_THD        40
#define KEY11_THD        40
#define KEY12_THD        40
#define KEY13_THD        40
#define KEY14_THD        40
#define KEY15_THD        40
#define KEY16_THD        40
#define KEY17_THD        40
#define KEY18_THD        40
#define KEY19_THD        40
#define KEY20_THD        40
#define KEY21_THD        40
#define KEY22_THD        40
#define KEY23_THD        40
#define KEY24_THD        40
#define KEY25_THD        40
```

1） CH_KEYxx 用于定义 KEYxx 对应的触摸通道，按键必须按顺序定义，最大可定义 26 个按键。没用到的按键通道要定义为 TK_CH_NONE。

2） KEYXX_THD 用于定义 CH_KEYxx 按键通道的门限值，当触摸差值超过 KEYXX_THD 时，按键触发。

5.2 触摸防水以及高灵敏度模式配置

```
//=============================================================
#define WATER_PROOF_EN          0           /*  1：防水功能使能   0：防水功能关闭   */
#define WATER_SHIELD_CH         TK_CH_NONE  /*  Shield通道号  */
#define WATER_PROOF_MODE        0           /*  0：有水状态下按键正常触发    1：有水状态下屏蔽按键 */
#define WATER_RATIO_0           15          /*  有水按键判断门限值   */
#define WATER_RATIO_1           45          /*  水流判断门限值       */
#define DUSTERCLOTH_EN          0           /*  1：防湿抹布功能开启      0：防湿抹布功能关闭*/
#define DUSTERCLOTH_THD         250         /*  湿抹布判断门限值 */

#define HIGH_SENSITVITY_EN      0           /*  1：高灵敏度模式开启  0：高灵敏度模式关闭 */
#define SENSITVITY_SHIELD_CH    TK_CH_NONE      /* Shield通道号 */
```

1） WATER_PROOF_EN　　用于开启或关闭防水功能；（1 为开启防水，0 为关闭防水）（开启防水功能后需开启输出补偿，并将补偿类似设置为同相）

2） WATER_SHIELD_CH　　用于设置 Shield 电极的触摸通道，未使用时填写 TK_CH_NONE。

3） WATER_PROOF_MODE　　用于设置防水模式；（0：有水状态下按键正常触发　　1：有水状态下屏蔽按键）。

4） DUSTERCLOTH_EN　　1：防湿抹布功能开启　　0：防湿抹布功能关闭

5） DUSTERCLOTH_THD　　用于设置湿抹布判断门限值

6） HIGH_SENSITVITY_EN，用于开启或关闭高灵敏模式；（1 为开启，0 为关闭）（开启防水功能后需开启输出补偿，并将补偿类似设置为同相）
7） SENSITVITY_SHIELD_CH 用于设置 Shield 电极的触摸通道，未使用时填写 TK_CH_NONE。

5.3 触摸滑条以及滑环配置

```
#define SLIDER_OR_WHEEL0_TYPE           TK_APP_NONE     //滑条类型
#define SLIDER_OR_WHEEL0_RESOLUTION     100               //滑条分辨率
#define SLIDER_OR_WHEEL0_THD            80                //滑条门限值
#define SLIDER_OR_WHEEL0_CH0            TK_CH4            //滑条通道，按顺序填写
#define SLIDER_OR_WHEEL0_CH1            TK_CH8
#define SLIDER_OR_WHEEL0_CH2            TK_CH5
#define SLIDER_OR_WHEEL0_CH3            TK_CH6
#define SLIDER_OR_WHEEL0_CH4            TK_CH7
#define SLIDER_OR_WHEEL0_CH5            TK_CH_NONE
#define SLIDER_OR_WHEEL0_CH6            TK_CH_NONE
#define SLIDER_OR_WHEEL0_CH7            TK_CH_NONE
```

1） SLIDER_OR_WHEELx_TYPE 用于定于滑条类型，可选择 TK_APP_SLIDER 或者 TK_APP_WHEEL，
TK_APP_WHEEL：表示为滑环，TK_APP_SLIDER：表示为滑条，当选择为其他时表示无效；
2） SLIDER_OR_WHEELx_RESOLUTION 用于定义滑条的分辨率；
3） SLIDER_OR_WHEELx_THD 用于定义滑条的触发门限值；
4） SLIDER_OR_WHEELx_CHx 用于定义滑条的通道，一个滑条最大支持 8 个通道；

PUYA CONFIDENTIAL

6. 常见烧录问题

1、出现如下错误提示后无法烧录

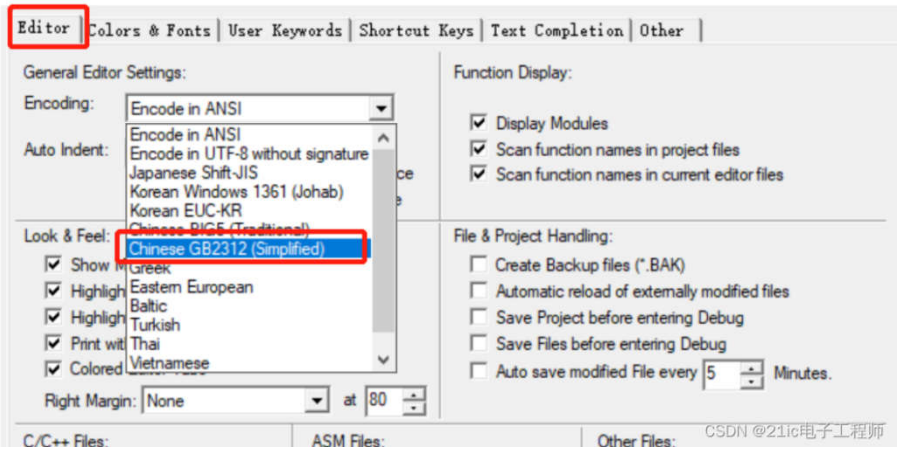
解决办法：出现这个提示一般是进入了休眠状态，需要唤醒后才能正常烧录；

2、SWD 口复用造成识别不对芯片

前一次下载的程序设置了 SWDIO/SWCLK 为 GPIO，导致 SWD 无法用于下载（**注：建议在 SWD 复用前加足够烧录下载的时间，等功能调试完成后可去除**）

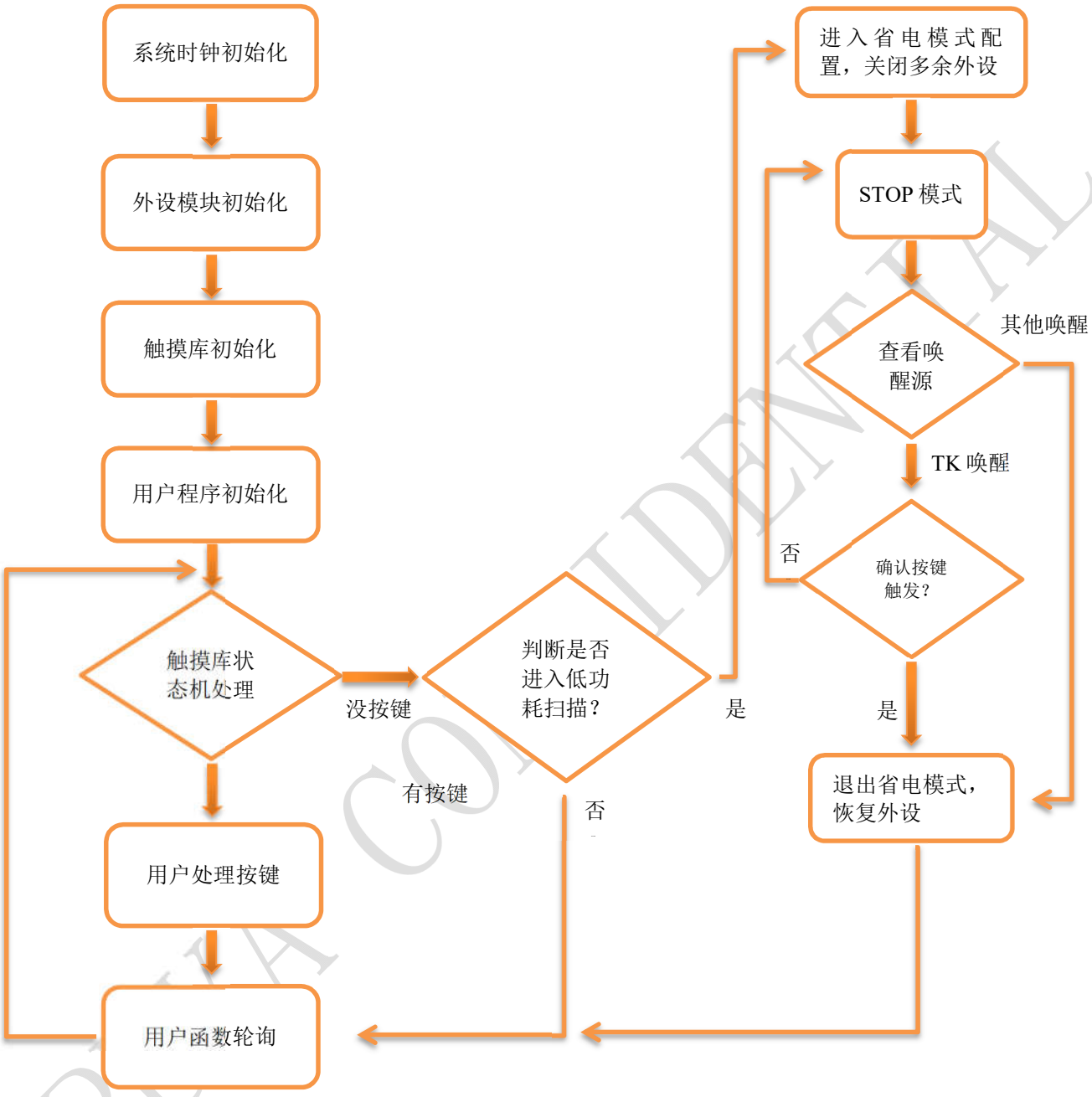
3、注释中的中文字体乱码

解决方法：在 Edit-configuration 中，Editor-Encoding 改为 Chinese GB2312 即可。需要将乱码删掉，重新输入就不会出现乱码了



PUYA CONFIDENTIAL

7. 程序流程图



PUYA CONFIDENTIAL

8. 更新历史

Version	Content	Date
V1.0	Initial version	2024/1/30
V1.2.0	增加滑条以及防水功能,修改触摸库结构	2024/3/27
V1.2.9	增加外设库，优化触摸库	2024/5/19

PUYA CONFIDENTIAL

Puya Semiconductor Co., Ltd.

声　　明

普冉半导体(上海)股份有限公司（以下简称：“Puya”）保留更改、纠正、增强、修改 Puya 产品和/或本文档的权利，恕不另行通知。用户可在下单前获取产品的最新相关信息。

Puya 产品是依据订单时的销售条款和条件进行销售的。

用户对 Puya 产品的选择和使用承担全责，同时若用于其自己或指定第三方产品上的，Puya 不提供服务支持且不对此类产品承担任何责任。

Puya 在此不授予任何知识产权的明示或暗示方式许可。

Puya 产品的转售，若其条款与此处规定不一致，Puya 对此类产品的任何保修承诺无效。

任何带有 Puya 或 Puya 标识的图形或字样是普冉的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代并替换先前版本中的信息。

普冉半导体(上海)股份有限公司 - 保留所有权利