SYD8811_BLE_Lib

盛芯微Confidential

AUTHOR

版本 Version 2.6.1

2020年 一月 15日 星期三

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

目录

SYDTEK Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

盛芯微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

模块索引

模块

这里列出了所有模块:

结构体索引

结构体

这里列出了所有结构体，并附带简要说明:

SYDTEK Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

文件索引

文件列表

这里列出了所有文件，并附带简要说明:

芯微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

模块说明

低功耗蓝牙协议栈控制接口

函数

- uint8_t BleInit (void)
  低功耗蓝牙初始化函数
- uint8_t DisConnect (void)
  主动断开蓝牙连接，
- uint8_t SetDevAddr (struct gap_ble_addr *p_dev_addr)
  设置蓝牙地址
- uint8_t GetDevAddr (struct gap_ble_addr *p_dev_addr)
  获取当前蓝牙地址
- uint8_t SetLEFeature (uint8_t *p_feature)
  设置蓝牙特性支持
- uint8_t SetAdvParams (struct gap_adv_params *p_adv_params)
  设置广播参数
- uint8_t SetAdvData (uint8_t *p_adv, uint8_t adv_sz, uint8_t *p_scan, uint8_t sacn_sz)
  设置广播数据和扫描回应数据
- uint8_t StartAdv (void)
  开始广播

  广播类型，广播间隔，广播周期等设置见接口SetAdvParams

  广播数据设置见接口SetAdvData
- uint8_t StopAdv (void)
  停止广播
- uint8_t SetScanParams (struct gap_scan_params *p_scan_params)
  扫描参数设置
- uint8_t StartScan (void)
  开始扫描
- uint8_t StopScan (void)
  停止扫描
- uint8_t SetSecParams (struct smp_pairing_req *p_sec_params)
  设置配对请求参数

  slave角色会在配对回应协议中发送， master角色会在配对请求协议中发送
- uint8_t SetPasskey (uint32_t passkey)
  设置passkey，在配对过程中根据连接双方的io能力确定

  是否需要输入或显示一个passkey，如果是justwork则无需设置
- uint8_t SecurityReq (uint8_t flag, uint8_t mitm)
  当slave主动要求发起配对时，调用该接口
- uint8_t SetConnectionUpdate (struct gap_update_params *p_update_params)
  当slave主动要求更新连接参数时，调用该接口
- uint8_t GetLinkParameters (struct gap_link_params *p_link)
  获取当前链路的连接参数
- uint8_t SetWinWideMinusCnt (uint8_t cnt)
- uint8_t ConnectionLatencyMode (uint8_t en)
  连接latency模式控制，latency模式请参考蓝牙Spec

芯微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

- uint8_t SetEvtCallback (struct gap_evt_callback *p_callback)
  设置GAP事件回调函数，当需要接收任何协议栈事件通知时， 都需要调用该接口来注册回调函数

  在协议栈事件发生时，会调用注册的 回调函数来通知对应的事件

  以事件回调方式来完成协议栈到应用程序的通信， 反过来应用程序到协议栈的通信方式为接口调用
- uint8_t GetGATTReportHandle (struct gap_att_report_handle **p_hdl)
  获取GATT所有的通知和指示信息列表
- uint8_t SetGATTReadRsp (uint8_t len, uint8_t *p_data)
  设置GATT需要读取的数据
- uint8_t CheckFIFOFull (void)
  检查协议栈FIFO是否已满
- uint8_t GATTDataSend (uint8_t type, struct gap_att_report *p_report, uint8_t len, uint8_t *p_data)
  gatt数据发送
- uint8_t Rand (void)
  随机数生成
- void DelayUS (uint16_t dly)
  延迟函数
- void DelayMS (uint32_t dly)
  延迟函数
- uint8_t GetCompanyID (void)
  获取公司ID
- uint8_t GetQFNType (void)
  获取芯片QFN 封装类型
- void RFRead (uint8_t addr, uint8_t *data)
- void RFWrite (uint8_t addr, uint8_t data)
- void ble_sched_execute (void)
  协议栈日程执行
- bool ble_sched_finish (void)
  检查协议栈日程是否执行完成

详细描述

函数说明

uint8_t BleInit (void)

低功耗蓝牙初始化函数

注解:

在使用蓝牙功能之前必须要调用该函数完成蓝牙初始化

返回:

初始化结果， _RETURN_STATUS_

盛芯微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

uint8_t DisConnect (void)

主动断开蓝牙连接，

注解:

请在蓝牙连接状态下调用该接口

返回:

接口调用结果， RETURN_STATUS

uint8_t SetDevAddr (struct gap_ble_addr * *p_dev_addr*)

设置蓝牙地址

注解:

该接口通常在初始化时设置一次

参数:

p_dev_addr	- 要设置的蓝牙地址类型和蓝牙地址

返回:

蓝牙地址设置是否成功， RETURN_STATUS

uint8_t GetDevAddr (struct gap_ble_addr * *p_dev_addr*)

获取当前蓝牙地址

参数:

p_dev_addr	- 获取的蓝牙地址类型和蓝牙地址

返回:

蓝牙地址获取是否成功， RETURN_STATUS

uint8_t SetLEFeature (uint8_t * *p_feature*)

设置蓝牙特性支持

注解:

该接口通常不需设置，请在明确芯片蓝牙功能和特性定义的情况下使用

参数:

p_feature	- 要设置的蓝牙特性，蓝牙特性定义为8byte， 详细请参考蓝牙Spec

返回:

特性设置是否成功， RETURN_STATUS

uint8_t SetAdvParams (struct gap_adv_params * *p_adv_params*)

设置广播参数

芯片微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

注解:

在开始广播之前需调用该接口，开始广播接口StartAdv

参数:

p_adv_params	- 要设置的广播参数

返回:

广播参数设置是否成功， _RETURN_STATUS_

uint8_t SetAdvData (uint8_t * *p_adv*, uint8_t *adv_sz*, uint8_t * *p_scan*, uint8_t *sacn_sz*)

设置广播数据和扫描回应数据

注解:

在开始广播之前需调用该接口，开始广播接口StartAdv

参数:

p_adv	- 要设置的广播数据，如果没有广播数据则为NULL
adv_sz	- 广播数据长度，如果没有广播数据则长度为0
p_scan	- 要设置的扫描回应数据，如果没有扫描回应数据则为NULL
sacn_sz	- 扫描回应数据长度，如果没有扫描回应数据则长度为0

返回:

数据设置是否成功， _RETURN_STATUS_

uint8_t StartAdv (void)

开始广播

广播类型，广播间隔，广播周期等设置见接口SetAdvParams

广播数据设置见接口SetAdvData

返回:

开始广播接口调用是否成功， _RETURN_STATUS_

uint8_t StopAdv (void)

停止广播

注解:

请在广播期间调用此接口来停止广播

返回:

停止广播接口调用是否成功， _RETURN_STATUS_

uint8_t SetScanParams (struct gap_scan_params * *p_scan_params*)

扫描参数设置

盛士微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

注解:

在扫描开始前需要先调用此接口设置扫描参数

参数:

p_scan_params	- 要设置的扫描参数

返回:

扫描参数设置是否成功， _RETURN_STATUS_

uint8_t StartScan (void)

开始扫描

注解:

扫描模式，扫描间隔，扫描窗口等设置见接口SetScanParams

返回:

开始扫描接口调用是否成功， _RETURN_STATUS_

uint8_t StopScan (void)

停止扫描

注解:

请在扫描中调用此接口来停止扫描

返回:

停止扫描接口调用是否成功， _RETURN_STATUS_

uint8_t SetSecParams (struct smp_pairing_req * *p_sec_params*)

设置配对请求参数

slave角色会在配对回应协议中发送， master角色会在配对请求协议中发送

注解:

该接口通常在初始化时设置一次

返回:

配对请求参数设置是否成功， _RETURN_STATUS_

uint8_t SetPasskey (uint32_t *passkey*)

设置passkey，在配对过程中根据连接双方的io能力确定

是否需要输入或显示一个passkey，如果是justwork则无需设置

注解:

该接口应当在GAP事件上报的情况下使用，上报事件:

GAP_EVT_PASSKEY_REQ

GAP_EVT_SHOW_PASSKEY_REQ

芯微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

参数:

passkey	- 要设置的passkey，该值是一个6位的数字，如: 123456

返回:

passkey设置是否成功， _RETURN_STATUS_

uint8_t SecurityReq (uint8_t *flag*, uint8_t *mitm*)

当slave主动要求发起配对时，调用该接口

注解:

通常情况下，当master收到该请求后会发起配对

参数:

flag	- 绑定标志，0x00: 不绑定，0x01: 绑定
mitm	- MITM保护，0x00: 不要求MITM保护，0x01: 要求MITM保护

返回:

安全请求是否成功， _RETURN_STATUS_

uint8_t SetConnectionUpdate (struct gap_update_params * *p_update_params*)

当slave主动要求更新连接参数时，调用该接口

注解:

当master收到连接参数更新请求后，会回复连接参数更新回应， 该回应以事件的形式通知，GAP_EVT_L2CAP_UPDATE_RSP

如果回应为接受参数请求，接下来master会发起连接更新， 更新完成后会收到事件通知，GAP_EVT_CONN_UPDATE_COMP

如果回应为拒绝参数请求，slave可调整参数后继续调用该接口

参数:

p_update_params	- 请求更新的连接参数

返回:

连接更新请求接口调用是否成功， _RETURN_STATUS_

uint8_t GetLinkParameters (struct gap_link_params * *p_link*)

获取当前链路的连接参数

参数:

p_link	- 获取的链路连接参数

返回:

获取链路连接参数是否成功， _RETURN_STATUS_

uint8_t SetWinWideMinusCnt (uint8_t *cnt*)

uint8_t ConnectionLatencyMode (uint8_t *en*)

盛士微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

连接latency模式控制，latency模式请参考蓝牙Spec

注解:

latency是降低slave功耗的一种方式，在考虑系统使用功耗时调用
通常在蓝牙数据处理完成后使能， 数据到来时失能

参数:

en	- latency使能控制，0：失能，1：使能

返回:

latency模式配置是否成功， RETURN_STATUS

uint8_t SetEvtCallback (struct gap_evt_callback * *p_callback*)

设置GAP事件回调函数，当需要接收任何协议栈事件通知时， 都需要调用该接口来注册回调函数

在协议栈事件发生时，会调用注册的 回调函数来通知对应的事件

以事件回调方式来完成协议栈到应用程序的通信，反过来应用程序到协议栈的通信方式为接口调用

注解:

该接口通常在初始化时调用一次
通过该注册接口可控制哪些事件需要上报，通常会屏蔽太过频繁的事件 来减少系统开销，所有可配置的事件参考 GAP_EVT
特别注意，回调函数可能在系统中断中调用，为减少中断耗时，请在 回调函数中做最少的处理，比如有大量数据时通过转存的方式到外部处理

参数:

p_callback	- 要注册的回调函数及参数

返回:

回调设置是否成功， RETURN_STATUS

uint8_t GetGATTReportHandle (struct gap_att_report_handle ** *p_hdl*)

获取GATT所有的通知和指示信息列表

参数:

p_hdl	- 获取的通知和指示信息

返回:

信息获取是否成功， RETURN_STATUS

uint8_t SetGATTReadRsp (uint8_t *len*, uint8_t * *p_data*)

设置GATT需要读取的数据

注解:

该接口在att读事件中调用，att读事件GAP_EVT_ATT_READ

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

盛达微Confidential

如果不调用该接口则回应默认值
当对应handle有att写操作时应当保存写的值，在att读事件时 通过该接口来设置之前保存的值

参数:

len	- att数据长度
p_data	- att数据

返回:

数据设置是否成功， _RETURN_STATUS_

uint8_t CheckFIFOFull (void)

检查协议栈FIFO是否已满

注解:

通常在需要发送gatt数据之前检测，若未满再发送
gatt数据发送接口GATTDataSend

返回:

0: FIFO未满
1: FITO已满

uint8_t GATTDataSend (uint8_t *type*, struct gap_att_report * *p_report*, uint8_t *len*, uint8_t * *p_data*)

gatt数据发送

注解:

slave角色中，gatt为server端，需要主动发送的数据只有通知和指示
如果需要发现client端的服务，请参考GATT Client端服务发现接口

参数:

type	- 发送数据类型BLE_SEND_TYPE
p_report	- 发送的特征信息
len	- 发送的特征值的长度
p_data	- 发送的特征值

返回:

调用发送接口是否成功， _RETURN_STATUS_

uint8_t Rand (void)

随机数生成

返回:

生成的随机数

void DelayUS (uint16_t *dly*)

延迟函数

芯片微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

参数:

dly	- 延迟微妙数

返回:

void

void DelayMS (uint32_t *dly*)

延迟函数

参数:

dly	- 延迟毫秒数

返回:

void

uint8_t GetCompanyID (void)

获取公司ID

返回:

公司ID ， 参考 COMPANY_ID

uint8_t GetQFNType (void)

获取芯片QFN 封装类型

返回:

QFN 类型，参考 QFN_TYPE

void RFRead (uint8_t *addr*, uint8_t * *data*)

void RFWrite (uint8_t *addr*, uint8_t *data*)

void ble_sched_execute (void)

协议栈日程执行

注解:

通常需要在应用程序里周期调用或 在while(1)里调用

返回:

void

bool ble_sched_finish (void)

检查协议栈日程是否执行完成

芯片微 Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

返回:

true：执行完成，false：未完成

系统时钟控制接口

函数

- uint8_t LPOCalibration (void)
  内部32k LPO时钟校准
- uint8_t RCOSCCalibration (void)
  内部RCOSC时钟校准
- uint8_t MCUClockSwitch (uint8_t sel)
  mcu时钟切换
- uint8_t ClockSwitch (uint8_t sel)
  32k 时钟切换
- uint8_t GetMCUClock (uint8_t *p_sel)
  获取当前mcu时钟
- uint8_t GetClock (uint8_t *p_sel)
  获取当前32k 时钟

详细描述

函数说明

uint8_t LPOCalibration (void)

内部32k LPO时钟校准

注解:

LPO时钟稳定度高，无需频繁校准，推荐5分钟校准一次

返回:

时钟校准是否成功， RETURN_STATUS

uint8_t RCOSCCalibration (void)

内部RCOSC时钟校准

注解:

RCOSC时钟稳定度高，无需频繁校准
推荐初始化校准一次

芯片微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

返回:

时钟校准是否成功， RETURN_STATUS

uint8_t MCUClockSwitch (uint8_t *sel*)

mcu时钟切换

参数:

sel	- 选择需要切换的时钟 SYSTEM_CLOCK_SEL

返回:

时钟切换是否成功， RETURN_STATUS

uint8_t ClockSwitch (uint8_t *sel*)

32k 时钟切换

参数:

sel	- 选择需要切换的时钟 32K_CLOCK_SEL

返回:

时钟切换是否成功， RETURN_STATUS

uint8_t GetMCUClock (uint8_t * *p_sel*)

获取当前mcu时钟

参数:

p_sel	- 获取的mcu时钟 SYSTEM_CLOCK_SEL

返回:

时钟获取是否成功， RETURN_STATUS

uint8_t GetClock (uint8_t * *p_sel*)

获取当前32k 时钟

参数:

p_sel	- 获取的32k时钟 32K_CLOCK_SEL

返回:

时钟获取是否成功， RETURN_STATUS

系统sleep及wakeup相关接口

函数

- uint8_t WakeupConfig (struct gap_wakeup_config *p_cfg)

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

芯片微 Confidential

系统唤醒配置接口

- uint8_t LLSleep (void)
- uint8_t SystemSleep (void)
  MCU进入sleep状态
- uint8_t SystemPowerDown (void)
  系统下电
- uint8_t SystemReset (void)
  系统软复位
- uint8_t RFSleep (void)
- uint8_t RFWakeup (void)
- uint8_t UartEn (uint8_t en)
  在RF sleep时，配置uart使能工作

详细描述

函数说明

uint8_t WakeupConfig (struct gap_wakeup_config * *p_cfg*)

系统唤醒配置接口

参数:

p_cfg	- 要配置的唤醒参数

返回:

参数配置是否成功， RETURN_STATUS

uint8_t LLSleep (void)

uint8_t SystemSleep (void)

MCU进入sleep状态

注解:

通常在应用程序处理完所有的事务后，需要进一步降低功耗时 可调用该接口
调用该接口后mcu立即进入sleep状态，直到配置的唤醒源唤醒后 mcu会继续sleep前的程序连续运行，并不会复位运行
调用该接口前需要先配置好唤醒源，WakeupConfig
蓝牙收发包会自动唤醒mcu

返回:

接口调用是否成功， RETURN_STATUS

uint8_t SystemPowerDown (void)

系统下电

盛达微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

注解:

调用该接口后，整个系统下电，仅RTC域有电，此时系统功耗降至最低
进入该模式后只能通过配置的唤醒源唤醒，唤醒后系统复位运行
调用该接口前需要先配置好唤醒源，WakeupConfig

返回:

接口调用是否成功，_RETURN_STATUS_

uint8_t SystemReset (void)

系统软复位

注解:

调用该接口后系统会复位运行

返回:

接口调用是否成功，_RETURN_STATUS_

uint8_t RFSleep (void)

uint8_t RFWakeup (void)

uint8_t UartEn (uint8_t *en*)

在RF sleep时，配置uart使能工作

注解:

在RF sleep时使能该接口uart才能正常工作
通常在程序中，若蓝牙和uart需要同时使用时， 可在初始化时调用该接口一次
注：使能该接口会引起系统功耗的增加

参数:

en	- 使能控制，01：使能，00：失能

返回:

配置是否成功，_RETURN_STATUS_

盛达微 Confidential

蓝牙设备绑定控制接口

函数

- uint8_t SetBondManagerIndex (uint8_t idx)
  设置当前bonding的索引位置
- uint8_t GetBondDevice (struct gap_bond_dev *p_device)
  获取当前bonding设备的信息
- uint8_t AddBondDevice (struct gap_bond_dev *p_device)
  增加bonding设备信息
- uint8_t DelBondDevice (void)

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

删除bonding设备信息

- uint8_t DelAllBondDevice (void)
  删除所有bonding设备信息

详细描述

函数说明

uint8_t SetBondManagerIndex (uint8_t *idx*)

设置当前bonding的索引位置

参数:

idx	- 要设置的索引位，系统支持最多10个绑定，取值范围：0~9

返回:

索引位设置是否成功， RETURN_STATUS

uint8_t GetBondDevice (struct gap_bond_dev * *p_device*)

获取当前bonding设备的信息

参数:

p_device	- 绑定设备的信息

返回:

获取结果是否成功， RETURN_STATUS

uint8_t AddBondDevice (struct gap_bond_dev * *p_device*)

增加bonding设备信息

参数:

p_device	- 要增加的bonding设备信息

返回:

增加bonding信息是否成功， RETURN_STATUS

uint8_t DelBondDevice (void)

删除bonding设备信息

注解:

设置要删除的bonding索引位接口SetBondManagerIndex

盛达微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

返回:

删除bonding信息是否成功， RETURN_STATUS

uint8_t DelAllBondDevice (void)

删除所有bonding设备信息

返回:

删除所有bonding信息是否成功， RETURN_STATUS

Profile读写flash数据接口

函数

- uint8_t ReadProfileData (uint16_t addr, uint16_t len, uint8_t *p_buf)
  读profile数据
- uint8_t WriteProfileData (uint16_t addr, uint16_t len, uint8_t *p_buf)
  写profile数据
- uint8_t EraseFlashData (uint32_t addr, uint8_t sector_num)
  擦出flash数据
- uint8_t ReadFlashData (uint32_t addr, uint16_t len, uint8_t *p_buf)
  读flash数据
- uint8_t WriteFlashData (uint32_t addr, uint16_t len, uint8_t *p_buf)
  写flash数据

详细描述

函数说明

uint8_t ReadProfileData (uint16_t *addr*, uint16_t *len*, uint8_t * *p_buf*)

读profile数据

注解:

协议栈在flash上预留了3k的数据存储区供profile存储数据使用

参数:

addr	- 读起始地址，范围：0~0xbff
len	- 读数据的长度，范围：1~0xc00
p_buf	- 读数据buffer

返回:

读数据是否成功， RETURN_STATUS

盛士微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

uint8_t WriteProfileData (uint16_t *addr*, uint16_t *len*, uint8_t * *p_buf*)

写profile数据

注解:

协议栈在flash上预留了3k的数据存储区供profile存储数据使用

参数:

addr	- 写起始地址，范围：0~0xbff
len	- 写数据的长度，范围：1~128
p_buf	- 写数据buffer

返回:

写数据是否成功， RETURN_STATUS

uint8_t EraseFlashData (uint32_t *addr*, uint8_t *sector_num*)

擦出flash数据

注解:

特殊使用，可通过修改配置来使用flash剩余的空间存储数据
注：此接口详细请咨询技术支持

参数:

addr	- 擦出起始地址，需在4k byte边界上
sector_num	- 擦出的sector数

返回:

擦出是否成功， RETURN_STATUS

uint8_t ReadFlashData (uint32_t *addr*, uint16_t *len*, uint8_t * *p_buf*)

读flash数据

注解:

特殊使用，可通过修改配置来使用flash剩余的空间存储数据
注：此接口详细请咨询技术支持

参数:

addr	- 读起始地址，需word对齐且在word边界上
len	- 读数据的长度，需word对齐且在word边界上
p_buf	- 读数据buffer

返回:

读数据是否成功， RETURN_STATUS

uint8_t WriteFlashData (uint32_t *addr*, uint16_t *len*, uint8_t * *p_buf*)

写flash数据

Confidential 保密

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

注解:

特殊使用，可通过修改配置来使用flash剩余的空间存储数据
注：此接口详细请咨询技术支持

参数:

addr	- 写起始地址，需word对齐且在word边界上
len	- 写数据的长度，需word对齐且在word边界上
p_buf	- 写数据buffer

返回:

写数据是否成功， _RETURN_STATUS_

Firmware OTA升级接口

函数

- uint8_t CodeErase (void)
  擦出Firmware code
- uint8_t CodeWrite (uint16_t offset, uint16_t len, uint8_t *p_buf)
  写Firmware code
- uint8_t CodeUpdate (uint8_t *p_desc, uint8_t *p_ver, uint16_t sz, uint16_t checksum)
  更新Firmware code 描述，检查checkum

详细描述

函数说明

uint8_t CodeErase (void)

擦出Firmware code

返回:

擦出是否成功， _RETURN_STATUS_

uint8_t CodeWrite (uint16_t *offset*, uint16_t *len*, uint8_t * *p_buf*)

写Firmware code

参数:

offset	- 写偏移地址，需word对齐且在word边界上
len	- 写数据的长度，需word对齐且在word边界上
p_buf	- 写数据buffer

返回:

写数据是否成功， _RETURN_STATUS_

盛芯微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

uint8_t CodeUpdate (uint8_t * *p_desc*, uint8_t * *p_ver*, uint16_t *sz*, uint16_t *checksum*)

更新Firmware code 描述，检查checkum

参数:

p_desc	- Firmware描述
p_ver	- Firmware版本号
sz	- Firmware code 长度
checksum	- 校验

返回:

Firmware code设置和校验是否成功， _RETURN_STATUS_

GATT Client端服务发现接口

函数

- uint8_t ATTCSetCallback (p_attc_callback pfn_callback)
  ATT Client设置回调函数接口
- uint8_t ATTCMTUReq (uint16_t mtu)
  ATT Client MTU Request
- uint8_t ATTCFindInfoReq (uint16_t start_hdl, uint16_t end_hdl)
  ATT Client Find Information Request
- uint8_t ATTCFindByTypeValueReq (uint16_t start_hdl, uint16_t end_hdl, uint16_t type, uint8_t val_sz, uint8_t *p_val)
  ATT Client Find By Type Value Request
- uint8_t ATTCReadByTypeReq (uint16_t start_hdl, uint16_t end_hdl, uint16_t type_sz, uint8_t *p_type)
  ATT Client Read By Type Request
- uint8_t ATTCReadReq (uint16_t hdl)
  ATT Client Read Request
- uint8_t ATTCReadBlobReq (uint16_t hdl, uint16_t offset)
  ATT Client Read Blob Request
- uint8_t ATTCReadMultipleReq (uint8_t hdl_sz, uint8_t *p_hdl)
  ATT Client Read Multiple Request
- uint8_t ATTCReadByGroupTypeReq (uint16_t start_hdl, uint16_t end_hdl, uint16_t type_sz, uint8_t *p_type)
  ATT Client Read By Group Type Request
- uint8_t ATTCWriteReq (uint16_t hdl, uint16_t sz, uint8_t *p_buf)
  ATT Client Write Request
- uint8_t ATTCWriteCmdReq (uint16_t hdl, uint16_t sz, uint8_t *p_buf)
  ATT Client Write Command Request
- uint8_t ATTCPrepareWriteReq (uint16_t hdl, uint16_t offset, uint16_t sz, uint8_t *p_buf)
  ATT Client Prepare Write Request
- uint8_t ATTCExecuteWriteReq (uint8_t flags)
  ATT Client Execute Write Request
- uint8_t ATTCConfirmation (void)
  ATT Client Confirmation

SYDTEK Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

详细描述

函数说明

uint8_t ATTCSetCallback (p_attc_callback *pfn_callback*)

ATT Client设置回调函数接口

参数:

in	*pfn_callback*	- 要设置的回调函数

返回:

设置回调函数是否成功， RETURN_STATUS

uint8_t ATTCMTUReq (uint16_t *mtu*)

ATT Client MTU Request

注解:

该指令只有在当前的GATT是处在client模式下才可使用，协议栈完全支持规范中l2cap层的功能，这里支持app 请求改变mtu，在需要传输大量数据的时候可以把MTU调大 当对方收到Exchange MTU Request后会相应发送Exchange MTU Response数据包， 当协议栈收到该数据包的时候，协议栈的GATT层会向gap_s_att_c_evt_handler_set函数指定的接口上报该事件

参数:

in	*mtu*	- 请求设置的mtu大小

返回:

指示 Exchange MTU Request指令是否发送成功， RETURN_STATUS

uint8_t ATTCFindInfoReq (uint16_t *start_hdl*, uint16_t *end_hdl*)

ATT Client Find Information Request

注解:

该指令只有在当前的GATT是处在client模式下才可使用，调用该接口协议栈将会发送Find Information Request命令，关于该命令的具体内容可看规范中ATT的相关章节 当对方收到att_c_findinforeq后会相应发送Find Information Response数据包，当协议栈收到该数据包的 时候，协议栈的GATT层会向gap_s_att_c_evt_handler_set函数指定的接口上报该事件

参数:

in	*start_hdl*	- 发送ATT的Find Information Request指令的Starting Handle参数
in	*end_hdl*	- 发送ATT的Find Information Request指令的Ending Handle参数

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

盛大微confidential

返回:

指示 Find Information Request指令是否发送成功， RETURN_STATUS

uint8_t ATTCFindByTypeValueReq (uint16_t *start_hdl*, uint16_t *end_hdl*, uint16_t *type*, uint8_t *val_sz*, uint8_t * *p_val*)

ATT Client Find By Type Value Request

注解:

该指令只有在当前的GATT是处在client模式下才可使用，调用该接口协议栈将会发送Find By Type Value Request命令，关于该命令的具体内容可看规范中ATT的相关章节 当对方收到Find By Type Value Request后会相应发送Find By Type Value Response数据包，当协议栈收到 该数据包的时候，协议栈的GATT层会向gap_s_att_c_evt_handler_set函数指定的接口上报该事件

参数:

in	*start_hdl*	- 发送ATT的Find By Type Value Request指令的Starting Handle参数
in	*end_hdl*	- 发送ATT的Find By Type Value Request指令的Ending Handle参数
in	*type*	- 发送ATT的Find By Type Value Request指令的Attribute Type参数
in	*val_sz*	- 发送ATT的Find By Type Value Request指令的Attribute Value参数的长度
in	*p_val*	- 发送ATT的Find By Type Value Request指令的Attribute Value参数

返回:

指示 Find By Type Value Request指令是否发送成功， RETURN_STATUS

uint8_t ATTCReadByTypeReq (uint16_t *start_hdl*, uint16_t *end_hdl*, uint16_t *type_sz*, uint8_t * *p_type*)

ATT Client Read By Type Request

注解:

该指令只有在当前的GATT是处在client模式下才可使用，调用该接口协议栈将会发送Read By Type Request 命令，关于该命令的具体内容可看规范中ATT的相关章节 当对方收到Read By Type Request后会相应发送Read By Type Response数据包，当协议栈收到该数据包的 时候，协议栈的GATT层会向gap_s_att_c_evt_handler_set函数指定的接口上报该事件

参数:

in	*start_hdl*	- 发送ATT的Read By Type Request指令的Starting Handle参数
in	*end_hdl*	- 发送ATT的Read By Type Request指令的Ending Handle参数
in	*type_sz*	- 发送ATT的Read By Type Request指令的Attribute Type参数的长度，根据规范这里只能够填充0或者16
in	*p_type*	- 发送ATT的Read By Type Request指令的Attribute Type参数

返回:

指示 Read By Type Request指令是否发送成功， RETURN_STATUS

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

Confidential 密级

uint8_t ATTCReadReq (uint16_t *hdl*)

ATT Client Read Request

注解:

该指令只有在当前的GATT是处在client模式下才可使用，调用该接口协议栈将会发送Read Request命令，关于 该命令的具体内容可看规范中ATT的相关章节 当对方收到Read Request后会相应发送Read Response数据包，当协议栈收到该数据包的时候协议栈的GATT层 会向gap_s_att_c_evt_handler_set函数指定的接口上报该事件

参数:

in	*hdl*	- 发送ATT的Read Request指令的Attribute Handle参数

返回:

指示Read Request指令是否发送成功， _RETURN_STATUS_

uint8_t ATTCReadBlobReq (uint16_t *hdl*, uint16_t *offset*)

ATT Client Read Blob Request

注解:

该指令只有在当前的GATT是处在client模式下才可使用，调用该接口协议栈将会发送Read Blob Request命令， 关于该命令的具体内容可看规范中ATT的相关章节 当对方收到 Read Blob Request后会相应发送Read Blob Response数据包，当协议栈收到该数据包的时候协议 栈的GATT层会向gap_s_att_c_evt_handler_set函数指定的接口上报该事件

参数:

in	*hdl*	- 发送ATT的Read Blob Request指令的Attribute Handle参数
in	*offset*	- 发送ATT的Read Blob Request指令的Value Offset参数

返回:

指示Read Blob Request指令是否发送成功， _RETURN_STATUS_

uint8_t ATTCReadMultipleReq (uint8_t *hdl_sz*, uint8_t * *p_hdl*)

ATT Client Read Multiple Request

注解:

该指令只有在当前的GATT是处在client模式下才可使用，调用该接口协议栈将会发送Read Multiple Request命令， 关于该命令的具体内容可看规范中ATT的相关章节 当对方收到 Read Multiple Request后会相应发送Read Multiple Response数据包，当协议栈收到该数据包的时候协议 栈的GATT层会向gap_s_att_c_evt_handler_set函数指定的接口上报该事件

参数:

in	*hdl_sz*	- 发送ATT的Read Multiple Request指令的Set Of Handles参数的长度，依据规范该参数不得小于4
in	*p_hdl*	- 发送ATT的Read Multiple Request指令的Set Of Handles参数

返回:

指示Read Multiple Request指令是否发送成功， _RETURN_STATUS_

盛大微 Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

uint8_t ATTCReadByGroupTypeReq (uint16_t *start_hdl*, uint16_t *end_hdl*, uint16_t *type_sz*, uint8_t * *p_type*)

ATT Client Read By Group Type Request

注解:

该指令只有在当前的GATT是处在client模式下才可使用，调用该接口协议栈将会发送Read by Group Type Request命令，关于该命令的具体内容可看规范中ATT的相关章节 当对方收到 Read by Group Type Request后会相应发送Read by Group Type Response数据包，当协议栈 收到该数据包的时候协议栈的GATT层会向gap_s_att_c_evt_handler_set函数指定的接口上报该事件

参数:

in	*start_hdl*	- 发送ATT的Read by Group Type Request指令的Starting Handle参数
in	*end_hdl*	- 发送ATT的Read by Group Type Request指令的Ending Handle参数
in	*type_sz*	- 发送ATT的Read by Group Type Request指令的Attribute Group Type参数的大小， 依据规范该大小必须为2或者16
in	*p_type*	- 发送ATT的Read by Group Type Request指令的Attribute Group Type参数

返回:

指示Read by Group Type Reques指令是否发送成功， _RETURN_STATUS_

uint8_t ATTCWriteReq (uint16_t *hdl*, uint16_t *sz*, uint8_t * *p_buf*)

ATT Client Write Request

注解:

该指令只有在当前的GATT是处在client模式下才可使用，调用该接口协议栈将会发送Read by Group Type Request命令，关于该命令的具体内容可看规范中ATT的相关章节 当对方收到 Read by Group Type Request后会相应发送Read by Group Type Response数据包，当协议栈 收到该数据包的时候协议栈的GATT层会向gap_s_att_c_evt_handler_set函数指定的接口上报该事件

参数:

in	*hdl*	- 发送ATT的Write Request指令的Attribute Handle参数
in	*sz*	- 发送ATT的Write Request指令的Attribute Value参数的大小
in	*p_buf*	- 发送ATT的Write Request指令的Attribute Value参数

返回:

指示Write Request指令是否发送成功， _RETURN_STATUS_

uint8_t ATTCWriteCmdReq (uint16_t *hdl*, uint16_t *sz*, uint8_t * *p_buf*)

ATT Client Write Command Request

设计微 Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

注解:

该指令只有在当前的GATT是处在client模式下才可使用，调用该接口协议栈将会发送Write Command命令，关 于该命令的具体内容可看规范中ATT的相关章节 因为Write Command命令并没有相应数据包，所以这里不会上报任何事件

参数:

in	*hdl*	- 发送ATT的Write Command指令的Attribute Handle参数
in	*sz*	- 发送ATT的Write Command指令的Attribute Value参数的大小
in	*p_buf*	- 发送ATT的Write Command指令的Attribute Value参数

返回:

指示Write Command指令是否发送成功， _RETURN_STATUS_

uint8_t ATTCPrepareWriteReq (uint16_t *hdl*, uint16_t *offset*, uint16_t *sz*, uint8_t * *p_buf*)

ATT Client Prepare Write Request

注解:

该指令只有在当前的GATT是处在client模式下才可使用，调用该接口协议栈将会发送Prepare Write Request 命令，关于该命令的具体内容可看规范中ATT的相关章节 当对方收到 Prepare Write Request后会相应发送Prepare Write Response数据包，当协议栈收到该数据包 的时候协议栈的GATT层会向gap_s_att_c_evt_handler_set函数指定的接口上报该事件

参数:

in	*hdl*	- 发送ATT的Prepare Write Request指令的Attribute Handle参数
in	*offset*	- 发送ATT的Prepare Write Request指令的Value Offset参数
in	*sz*	- 发送ATT的Prepare Write Request指令的Part Attribute Value参数的大小
in	*p_buf*	- 发送ATT的Write Command指令的Part Attribute Value参数

返回:

指示Prepare Write Request指令是否发送成功， _RETURN_STATUS_

uint8_t ATTCExecuteWriteReq (uint8_t *flags*)

ATT Client Execute Write Request

注解:

该指令只有在当前的GATT是处在client模式下才可使用，调用该接口协议栈将会发送Execute Write Request命令，关于该命令的具体内容可看规范中ATT的相关章节 当对方收到 Execute Write Request后会相应发送Execute Write Response数据包，当协议栈收到该数据包 的时候协议栈的GATT层会向gap_s_att_c_evt_handler_set函数指定的接口上报该事件

参数:

in	*flags*	- 发送ATT的Execute Write Request指令的Flags参数, ATT_EXEC_WRITE_FLAGS

返回:

指示Execute Write Request指令是否发送成功， _RETURN_STATUS_

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

盛天微Confidential

uint8_t ATTCConfirmation (void)

ATT Client Confirmation

注解:

该指令只有在当前的GATT是处在client模式下才可使用，调用该接口协议栈将会发送Handle Value Confirmation命令，关于该命令的具体内容可看规范中ATT的相关章节 因为Confirmation命令并没有相应数据包，所以这里不会上报任何事件

返回:

指示Confirmation指令是否发送成功

结构体说明

att_err_rsp结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint8_t opcode
  ATT错误响应中的Request Opcode In Error参数，ATT_CMD_CODE
- uint16_t hdl
  ATT错误响应中的Attribute Handle In Error参数
- uint8_t err
  ATT错误响应中的Error Code，ATT_ERROR_CODE

详细描述

BLE在发现请求条件出错的时候回应Error Response事件， 该ATT命令中包含有错误码和handle

结构体成员变量说明

uint8_t att_err_rsp::opcode

ATT错误响应中的Request Opcode In Error参数，ATT_CMD_CODE

uint16_t att_err_rsp::hdl

ATT错误响应中的Attribute Handle In Error参数

uint8_t att_err_rsp::err

ATT错误响应中的Error Code，ATT_ERROR_CODE

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

芯片微Confidential

att_find_by_type_val_rsp结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint8_t list [MAX_ATT_DATA_SZ-1]
  Handles Information List参数

详细描述

本结构体对应着Find By Type Value Response的参数，根据规范list是一个 Found Attribute Handle和Group End Handle的组合 该结构体用于GATT客户端，对于GATT的角色为服务器，那么该结构体是无用的。这里只提供ATT客户 端相应的API，并没有提供GAP的处理流程，以留给APP最大的灵活性

结构体成员变量说明

uint8_t att_find_by_type_val_rsp::list[MAX_ATT_DATA_SZ-1]

Handles Information List参数

att_find_info_128结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint16_t hdl
  Handle值
- uint8_t uuid [16]
  128bit的UUID值

详细描述

Find Information Response命令的format为0x02时，其Information Data为本结构体

芯片微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

结构体成员变量说明

uint16_t att_find_info_128::hdl

Handle值

uint8_t att_find_info_128::uuid[16]

128bit的UUID值

---

att_find_info_16结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint16_t hdl
  Handle值
- uint8_t uuid [2]
  16bit的UUID值

详细描述

Find Information Response命令的format为0x01时，其Information Data为本结构体

结构体成员变量说明

uint16_t att_find_info_16::hdl

Handle值

uint8_t att_find_info_16::uuid[2]

16bit的UUID值

---

att_find_info_payload联合体 参考

```
#include <lib_cn.h>
```

芯讯微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

成员变量

- struct att_find_info_16 uuid16 [5]
  16bituuid 下的Information Data参数
- struct att_find_info_128 uuid128
  128bituuid 下的Information Data参数

详细描述

本联合体对应着Find Information Response的Information Data参数，根据规范因为format不一样 Information Data有可能是handle和16bit uuid的组合，也有可能是handle和128bit uuid的组合

结构体成员变量说明

struct att_find_info_16 att_find_info_payload::uuid16[5]

16bituuid下的Information Data参数

struct att_find_info_128 att_find_info_payload::uuid128

128bituuid下的Information Data参数

---

att_find_info_rsp结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint8_t format
  *该参数决定了Information Data的长度，_GATT_FIND_INFO_UUID_TYPE_*
- union att_find_info_payload pair
  Information Data参数

详细描述

该结构体用于GATT客户端，对于GATT的角色为服务器，那么该结构体是无用的。这里只提供ATT客户 端相应的API，并没有提供GAP的处理流程，以留给APP最大的灵活性

Confidential 保密

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

结构体成员变量说明

uint8_t att_find_info_rsp::format

该参数决定了Information Data的长度， _GATT_FIND_INFO_UUID_TYPE_

union att_find_info_payload att_find_info_rsp::pair

Information Data参数

---

att_hdl_val_indication结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint16_t hdl
- uint8_t buf [MAX_ATT_DATA_SZ-3]

详细描述

本结构体对应譎andle Value Indication的参数，根据规范buf最大值为ATT_MTU 该结构体用于GATT客户端，对于GATT的角色为服务器，那么该结构体是无用的。这里只提供ATT客户 端相应的API，并没有提供GAP的处理流程，以留给APP最大的灵活性

结构体成员变量说明

uint16_t att_hdl_val_indication::hdl

Attribute Handle参数

uint8_t att_hdl_val_indication::buf[MAX_ATT_DATA_SZ-3]

Attribute Value参数

---

att_hdl_val_notifivation结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint16_t hdl
- uint8_t buf [MAX_ATT_DATA_SZ-3]

盛达微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

详细描述

本结构体对应着Handle Value Notification的参数，根据规范buf最大值为ATT_MTU 该结构体用于GATT客户端，对于GATT的角色为服务器，那么该结构体是无用的。这里只提供ATT客户 端相应的API，并没有提供GAP的处理流程，以留给APP最大的灵活性

结构体成员变量说明

uint16_t att_hdl_val_notifivation::hdl

Attribute Handle参数

uint8_t att_hdl_val_notifivation::buf[MAX_ATT_DATA_SZ-3]

Attribute Value参数

att_mtu_rsp结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint16_t mtu
  ATT层的Exchange MTU Response命令的Server Rx MTU参数

详细描述

本结构体对应着Exchange MTU Response的各个参数 该结构体用于GATT客户端，对于GATT的角色为服务器，那么该结构体是无用的。这里只提供ATT客户 端相应的API，并没有提供GAP的处理流程，以留给APP最大的灵活性

结构体成员变量说明

uint16_t att_mtu_rsp::mtu

ATT层的Exchange MTU Response命令的Server Rx MTU参数

att_read_blob_rsp结构体 参考

```
#include <lib_cn.h>
```


All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

成员变量

- uint8_t buf [MAX_ATT_DATA_SZ-1]
  Attribute Value参数

详细描述

该结构体用于GATT客户端，对于GATT的角色为服务器，那么该结构体是无用的。这里只提供ATT客户 端相应的API，并没有提供GAP的处理流程，以留给APP最大的灵活性

结构体成员变量说明

uint8_t att_read_blob_rsp::buf[MAX_ATT_DATA_SZ-1]

Attribute Value参数

---

att_read_by_group_type_128结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint16_t hdl
- uint16_t end_hdl
- uint8_t uuid [16]

详细描述

本结构体对应着Read by Group Type Response的参数

结构体成员变量说明

uint16_t att_read_by_group_type_128::hdl

128bituuid下的Attribute Data List的Attribute Handle参数

uint16_t att_read_by_group_type_128::end_hdl

128bituuid下的Attribute Data List的End Group Handle参数

uint8_t att_read_by_group_type_128::uuid[16]

128bituuid下的Attribute Data List的Attribute Value参数

盛士微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

att_read_by_group_type_16结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint16_t hdl
- uint16_t end_hdl
- uint8_t uuid [2]

详细描述

本结构体对应着Read by Group Type Response的参数

结构体成员变量说明

uint16_t att_read_by_group_type_16::hdl

16bituuid下的Attribute Data List的Attribute Handle参数

uint16_t att_read_by_group_type_16::end_hdl

16bituuid下的Attribute Data List的End Group Handle参数

uint8_t att_read_by_group_type_16::uuid[2]

16bituuid下的Attribute Data List的Attribute Value参数

att_read_by_group_type_payload联合体 参考

```
#include <lib_cn.h>
```

成员变量

- struct att_read_by_group_type_16 uuid16 [3]
- struct att_read_by_group_type_128 uuid128

详细描述

本结构体对应着Read by Group Type Response的参数，根据规范因为length不一样Attribute Data List 有可能是handle和16bit uuid的主要服务组合，也有可能是handle和128bit uuid的主要服务组合

芯片微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

结构体成员变量说明

struct att_read_by_group_type_16 att_read_by_group_type_payload::uuid16[3]

16bituuid下的Attribute Data List参数

struct att_read_by_group_type_128 att_read_by_group_type_payload::uuid128

128bituuid下的Attribute Data List参数

---

att_read_by_group_type_rsp结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint8_t length
- union att_read_by_group_type_payload pair

详细描述

本结构体对应着Read by Group Type Response的参数，根据规范因为length不一样Attribute Data List 有可能是handle和16bit uuid的主要服务组合，也有可能是handle和128bit uuid的主要服务组合 该结构体用于GATT客户端，对于GATT的角色为服务器，那么该结构体是无用的。这里只提供ATT客户 端相应的API，并没有提供GAP的处理流程，以留给APP最大的灵活性

结构体成员变量说明

uint8_t att_read_by_group_type_rsp::length

Length参数

union att_read_by_group_type_payload att_read_by_group_type_rsp::pair

Attribute Data List参数

---

att_read_by_type_128结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint16_t hdl
- uint8_t property
- uint16_t val_hdl

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

盛大微Confidential

- uint8_t char_uuid [16]

详细描述

本结构体对应着Read By Type Response参数中Attribute Data List的参数

结构体成员变量说明

uint16_t att_read_by_type_128::hdl

Attribute Data List参数中的Attribute Handle参数

uint8_t att_read_by_type_128::property

Attribute Data List参数中的Attribute Value参数，在发现characteristic时 Attribute Value中的Characteristic Properties

uint16_t att_read_by_type_128::val_hdl

Attribute Data List参数中的Attribute Value参数在发现characteristic时Attribute Value中的Characteristic Value Attribute Handle

uint8_t att_read_by_type_128::char_uuid[16]

Attribute Data List参数中的Attribute Value参数在发现characteristic时Attribute Value中的Characteristic UUID

---

att_read_by_type_16结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint16_t hdl
- uint8_t property
- uint16_t val_hdl
- uint8_t char_uuid [2]

详细描述

本结构体对应着Read By Type Response参数中Attribute Data List的参数

结构体成员变量说明

uint16_t att_read_by_type_16::hdl

Attribute Data List参数中的Attribute Handle参数

芯微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

uint8_t att_read_by_type_16::property

Attribute Data List参数中的Attribute Value参数，在发现characteristic时 Attribute Value中的Characteristic Properties

uint16_t att_read_by_type_16::val_hdl

Attribute Data List参数中的Attribute Value参数在发现characteristic时Attribute Value中的Characteristic Value Attribute Handle

uint8_t att_read_by_type_16::char_uuid[2]

Attribute Data List参数中的Attribute Value参数在发现characteristic时Attribute Value中的Characteristic UUID

---

att_read_by_type_chartextend_rsp结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint8_t length
  Length参数
- uint16_t hdl
  Attribute Data List参数中的Attribute Handle
- uint8_t val [MAX_ATT_DATA_SZ-4]
  128bituuid下的Attribute Data List参数

详细描述

本结构体对应着Read By Type Response的参数，根据规范因为length代表Attribute Data List的长度 本结构只是用于操作对象是特性，和att_read_by_type_rsp不同，这是一个扩展的结构体 该结构体用于GATT客户端，对于GATT的角色为服务器，那么该结构体是无用的。这里只提供ATT客户 端相应的API，并没有提供GAP的处理流程，以留给APP最大的灵活性

结构体成员变量说明

uint8_t att_read_by_type_chartextend_rsp::length

Length参数

uint16_t att_read_by_type_chartextend_rsp::hdl

Attribute Data List参数中的Attribute Handle

盛炬微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

uint8_t att_read_by_type_chartextend_rsp::val[MAX_ATT_DATA_SZ-4]

128bituuid下的Attribute Data List参数

---

att_read_by_type_include_rsp结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint8_t length
- uint16_t hdl
- uint8_t buf [MAX_ATT_DATA_SZ-2]

详细描述

本结构体对应着Read By Type Response的参数，根据规范因为length不一样，其中length决定了 Attribute Data List的长度 本结构只是用于操作对象是次要服务（include）的情况 该结构体用于GATT客户端，对于GATT的角色为服务器，那么该结构体是无用的。这里只提供ATT客户 端相应的API，并没有提供GAP的处理流程，以留给APP最大的灵活性

结构体成员变量说明

uint8_t att_read_by_type_include_rsp::length

ATT层的Read By Type Response命令下的Length参数

uint16_t att_read_by_type_include_rsp::hdl

ATT层的Read By Type Response命令下的Attribute Data List参数中的Attribute Handle

uint8_t att_read_by_type_include_rsp::buf[MAX_ATT_DATA_SZ-2]

ATT层的Read By Type Response命令下的Attribute Data List参数中的Attribute Value

---

att_read_by_type_pair_val结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint16_t hdl
  Attribute Data List参数中的Attribute Handle

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

芯片微Confidential

- uint8_t val [10]
  Attribute Data List参数中的Attribute Value

详细描述

本结构只是用于操作对象是单个pair值的情况

结构体成员变量说明

uint16_t att_read_by_type_pair_val::hdl

Attribute Data List参数中的Attribute Handle

uint8_t att_read_by_type_pair_val::val[10]

Attribute Data List参数中的Attribute Value

---

att_read_by_type_payload联合体 参考

```
#include <lib_cn.h>
```

成员变量

- struct att_read_by_type_16 uuid16 [3]
- struct att_read_by_type_128 uuid128

详细描述

Read By Type Response中的Length决定了Attribute Data List的长度，区别16bit uuid和1258bit uuid 本结构只是用于操作对象是characteristic的情况

结构体成员变量说明

struct att_read_by_type_16 att_read_by_type_payload::uuid16[3]

16bituuid下的Attribute Data List参数

struct att_read_by_type_128 att_read_by_type_payload::uuid128

128bituuid下的Attribute Data List参数

盛士微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

att_read_by_type_rsp结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint8_t length
- union att_read_by_type_payload pair

详细描述

本结构体对应着Read By Type Response的参数，根据规范因为length不一样，其中length决定了 Attribute Data List的长度 Attribute Data List有可能是handle和16bit uuid的characteristic组合，也有可能是handle和128bit uuid的 characteristic组合 本结构只是用于操作对象是characteristic的情况 该结构体用于GATT客户端，对于GATT的角色为服务器，那么该结构体是无用的。这里只提供ATT客户 端相应的API，并没有提供GAP的处理流程，以留给APP最大的灵活性

结构体成员变量说明

uint8_t att_read_by_type_rsp::length

ATT层的Read By Type Response命令下的Length参数

union att_read_by_type_payload att_read_by_type_rsp::pair

ATT层的Read By Type Response命令下的Attribute Data List参数

---

att_read_by_type_service_128结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint16_t hdl
  Attribute Data List参数中的Attribute Handle
- uint8_t uuid [16]
  Attribute Data List参数中的Attribute Value

详细描述

本结构体对应着Read By Type Response的参数 本结构只是用于操作对象是主要服务的情况

成片微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

结构体成员变量说明

uint16_t att_read_by_type_service_128::hdl

Attribute Data List参数中的Attribute Handle

uint8_t att_read_by_type_service_128::uuid[16]

Attribute Data List参数中的Attribute Value

---

att_read_by_type_service_16结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint16_t hdl
  Attribute Data List参数中的Attribute Handle
- uint8_t uuid [2]
  Attribute Data List参数中的Attribute Value

详细描述

本结构体对应着Read By Type Response的参数 本结构只是用于操作对象是主要服务的情况

结构体成员变量说明

uint16_t att_read_by_type_service_16::hdl

Attribute Data List参数中的Attribute Handle

uint8_t att_read_by_type_service_16::uuid[2]

Attribute Data List参数中的Attribute Value

---

att_read_by_type_service_payload联合体 参考

```
#include <lib_cn.h>
```

成士微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

成员变量

- struct att_read_by_type_service_16 uuid16 [3]
- struct att_read_by_type_service_128 uuid128

详细描述

本结构体对应着Read By Type Response的参数，根据规范因为length不一样Attribute Data List有可 能是handle和16bit uuid的主要服务组合，也有可能是handle和128bit uuid的主要服务组合 本结构只是用于操作对象是主要服务的情况

结构体成员变量说明

struct att_read_by_type_service_16 att_read_by_type_service_payload::uuid16[3]

16bituuid下的Attribute Data List参数

struct att_read_by_type_service_128 att_read_by_type_service_payload::uuid128

128bituuid下的Attribute Data List参数

---

att_read_by_type_service_rsp结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint8_t length
  Length参数
- union att_read_by_type_service_payload pair
  attribute Data List参数

详细描述

本结构体对应着Read By Type Response的参数，根据规范因为length不一样Attribute Data List有可 能是handle和16bit uuid的主要服务组合，也有可能是handle和128bit uuid的主要服务组合 本结构只是用于操作对象是主要服务的情况 该结构体用于GATT客户端，对于GATT的角色为服务器，那么该结构体是无用的。SYD8821只是提供ATT客户 端相应的API，并没有提供GAP的处理流程，以留给APP最大的灵活性

结构体成员变量说明

uint8_t att_read_by_type_service_rsp::length

Length参数

芯微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

union att_read_by_type_service_payload att_read_by_type_service_rsp::pair

attribute Data List参数

---

att_read_by_type_val_rsp结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint8_t length
  Length参数
- struct att_read_by_type_pair_val pair [1]
  Attribute Data List参数

详细描述

本结构体对应着Read By Type Response的参数，根据规范因为length不一样，其中length决定了 Attribute Data List的长度 本结构只是用于操作对象是单个pair值的情况 该结构体用于GATT客户端，对于GATT的角色为服务器，那么该结构体是无用的。这里只提供ATT客户 端相应的API，并没有提供GAP的处理流程，以留给APP最大的灵活性

结构体成员变量说明

uint8_t att_read_by_type_val_rsp::length

Length参数

struct att_read_by_type_pair_val att_read_by_type_val_rsp::pair[1]

Attribute Data List参数

---

att_read_multiple_rsp结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint8_t val [MAX_ATT_DATA_SZ-1]
  Set Of Values参数

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

盛大微 Confidential

详细描述

该结构体用于GATT客户端，对于GATT的角色为服务器，那么该结构体是无用的。这里只提供ATT客户 端相应的API，并没有提供GAP的处理流程，以留给APP最大的灵活性

结构体成员变量说明

uint8_t att_read_multiple_rsp::val[MAX_ATT_DATA_SZ-1]

Set Of Values参数

---

att_read_rsp结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint8_t buf [MAX_ATT_DATA_SZ-1]
  Attribute Value参数

详细描述

该结构体用于GATT客户端，对于GATT的角色为服务器，那么该结构体是无用的。这里只提供ATT客户 端相应的API，并没有提供GAP的处理流程，以留给APP最大的灵活性

结构体成员变量说明

uint8_t att_read_rsp::buf[MAX_ATT_DATA_SZ-1]

Attribute Value参数

---

attc_ble_evt结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint8_t attc_code
- uint8_t attc_sz

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

盛大微Confidential

- union {
- struct att_err_rsp AttErrRsp
- struct att_mtu_rsp AttMtuRsp
- struct att_find_info_rsp AttFindInfoRsp
- struct att_find_by_type_val_rsp AttFindByTypeValRsp
- struct att_read_by_type_rsp AttReadByTypeRsp
- struct att_read_by_type_include_rsp AttReadByTypeIncludeRsp
- struct att_read_by_type_val_rsp AttReadByTypeValRsp
- struct att_read_by_type_service_rsp AttReadByTypeServiceRsp
- struct att_read_by_type_chartextend_rsp AttReadByTypeChartExtendRsp
- struct att_read_rsp AttReadRsp
- struct att_read_blob_rsp AttReadBlobRsp
- struct att_read_multiple_rsp AttReadMultipleRsp
- struct att_read_by_group_type_rsp AttReadByGroupTypeRsp
- struct att_hdl_val_notifivation AttHdlValNotification
- struct att_hdl_val_indication AttHdlValIndication
- } attc

详细描述

收到蓝牙ATT客户端的事件的时候通过gap_s_att_c_evt_handler_set设置的接口上报蓝牙事件， attc_ble_evt结构体为蓝ATT客户端牙事件的具体数据 该结构体用于GATT客户端，对于GATT的角色为服务器，那么该结构体是无用的。这里只提供ATT客户 端相应的API，并没有提供GAP的处理流程，以留给APP最大的灵活性

结构体成员变量说明

uint8_t attc_ble_evt::attc_code

协议栈上报att客户端事件的操作码(Opcode)，ATT_CMD_CODE

uint8_t attc_ble_evt::attc_sz

协议栈上报att客户端事件的长度

struct att_err_rsp attc_ble_evt::AttErrRsp

ATT客户端错误事件的数据

struct att_mtu_rsp attc_ble_evt::AttMtuRsp

ATT客户端交换MTU事件的数据

struct att_find_info_rsp attc_ble_evt::AttFindInfoRsp

ATT客户端上报Find Information Response事件的数据

struct att_find_by_type_val_rsp attc_ble_evt::AttFindByTypeValRsp

ATT客户端上报Find By Type Response事件的数据

struct att_read_by_type_rsp attc_ble_evt::AttReadByTypeRsp

ATT客户端上报Read By Type Response事件的数据，该数据只适用于characteristic

Confidential 保密

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

struct att_read_by_type_include_rsp attc_ble_evt::AttReadByTypeIncludeRsp

ATT客户端上报Read By Type Response事件的数据，该数据只适用于包含服务

struct att_read_by_type_val_rsp attc_ble_evt::AttReadByTypeValRsp

ATT客户端上报Read By Type Response事件的数据，该数据只适用于单个pair值

struct att_read_by_type_service_rsp attc_ble_evt::AttReadByTypeServiceRsp

ATT客户端上报Read By Type Response事件的数据，该数据只适用于服务

struct att_read_by_type_chartextend_rsp attc_ble_evt::AttReadByTypeChartExtendRsp

ATT客户端上报Read By Type Response事件的数据，该数据只适用于扩展characteristic

struct att_read_rsp attc_ble_evt::AttReadRsp

ATT客户端上报Read Response事件的数据

struct att_read_blob_rsp attc_ble_evt::AttReadBlobRsp

ATT客户端上报Read Blob Response事件的数据

struct att_read_multiple_rsp attc_ble_evt::AttReadMultipleRsp

ATT客户端上报Read Multiple Response事件的数据

struct att_read_by_group_type_rsp attc_ble_evt::AttReadByGroupTypeRsp

ATT客户端上报Read by Group Type Response事件的数据

struct att_hdl_val_notifivation attc_ble_evt::AttHdlValNotification

ATT客户端上报Handle Value Notification事件的数据

struct att_hdl_val_indication attc_ble_evt::AttHdlValIndication

ATT客户端上报Handle Value Indication事件的数据

union { ... } attc_ble_evt::attc

盛天微Confidential

---

gap_adv_params结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint8_t type
  广播类型，参考 ADV_CH_PKT_TYPE
- struct gap_ble_addr peer_addr
  对端设备地址，如果广播类型为定向广播则必须设置
- uint8_t policy

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

过滤策略bitmap，对应bit设置1为使能，bit0：广播过滤，bit1：连接过滤

- uint8_t channel
  广播通道bitmap，对应bit设置1为使能，bit0: ch37, bit1: ch38, bit2: ch39
- uint16_t interval
  广播间隔，单位：0.625ms，取值范围0x0020~0x4000，即(20ms to 10.24s)
- uint16_t timeout
  广播周期，单位：1s，取值范围0x0001~0x3FFF，当设置为0x0000时只做单次广播

详细描述

广播参数设置，在开始广播之前必须要设置该参数

结构体成员变量说明

uint8_t gap_adv_params::type

广播类型，参考 ADV_CH_PKT_TYPE

struct gap_ble_addr gap_adv_params::peer_addr

对端设备地址，如果广播类型为定向广播则必须设置

uint8_t gap_adv_params::policy

过滤策略bitmap，对应bit设置1为使能，bit0：广播过滤，bit1：连接过滤

uint8_t gap_adv_params::channel

广播通道bitmap，对应bit设置1为使能，bit0: ch37, bit1: ch38, bit2: ch39

uint16_t gap_adv_params::interval

广播间隔，单位：0.625ms，取值范围0x0020~0x4000，即(20ms to 10.24s)

uint16_t gap_adv_params::timeout

广播周期，单位：1s，取值范围0x0001~0x3FFF，当设置为0x0000时只做单次广播

---

gap_att_exec_write_evt结构体 参考

```
#include <lib_cn.h>
```

芯片微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

成员变量

- uint8_t flags
  执行写标志，0：取消所有的预备写，1：立即写所有挂起的预备写

详细描述

att执行写事件

结构体成员变量说明

uint8_t gap_att_exec_write_evt::flags

执行写标志，0：取消所有的预备写，1：立即写所有挂起的预备写

---

gap_att_handle_configure_evt结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint16_t uuid
  特征UUID
- uint16_t hdl
  CCC handle
- uint16_t value
  CCC的值

详细描述

att客服端特征配置(CCC)事件

结构体成员变量说明

uint16_t gap_att_handle_configure_evt::uuid

特征UUID

uint16_t gap_att_handle_configure_evt::hdl

CCC handle

芯片微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

uint16_t gap_att_handle_configure_evt::value

CCC的值

---

gap_att_pre_write_evt结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint16_t primary
  写handle的主要服务
- uint16_t uuid
  特征UUID
- uint16_t hdl
  写handle
- uint16_t offset
  写偏移
- uint8_t sz
  写数据长度
- uint8_t data [MAX_ATT_DATA_SZ]
  写数据

详细描述

att预备写事件

结构体成员变量说明

uint16_t gap_att_pre_write_evt::primary

写handle的主要服务

uint16_t gap_att_pre_write_evt::uuid

特征UUID

uint16_t gap_att_pre_write_evt::hdl

写handle

uint16_t gap_att_pre_write_evt::offset

盛世微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

写偏移

uint8_t gap_att_pre_write_evt::sz

写数据长度

uint8_t gap_att_pre_write_evt::data[MAX_ATT_DATA_SZ]

写数据

---

gap_att_read_evt结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint16_t primary
  读handle的主要服务
- uint16_t uuid
  特征的UUID
- uint16_t hdl
  handle
- uint16_t offset
  读偏移

详细描述

att读事件

结构体成员变量说明

uint16_t gap_att_read_evt::primary

读handle的主要服务

uint16_t gap_att_read_evt::uuid

特征的UUID

uint16_t gap_att_read_evt::hdl

handle

盛士微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

uint16_t gap_att_read_evt::offset

读偏移

---

gap_att_report结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint16_t primary
  主要服务
- uint16_t uuid
  特征UUID
- uint16_t hdl
  特征handle
- uint16_t config
  通知或指示的配置handle
- uint16_t value
  通知或指示的配置值，0x0001：通知，0x0002：指示

详细描述

att通知和指示

结构体成员变量说明

uint16_t gap_att_report::primary

主要服务

uint16_t gap_att_report::uuid

特征UUID

uint16_t gap_att_report::hdl

特征handle

uint16_t gap_att_report::config

通知或指示的配置handle

芯片微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

uint16_t gap_att_report::value

通知或指示的配置值，0x0001：通知，0x0002：指示

---

gap_att_report_handle结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint8_t cnt
- struct gap_att_report report [MAX_ATT_REPORT_HDL]

详细描述

att所有通知和指示列表

结构体成员变量说明

uint8_t gap_att_report_handle::cnt

struct gap_att_report gap_att_report_handle::report[MAX_ATT_REPORT_HDL]

---

gap_att_write_evt结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint16_t primary
  写handle的主要服务
- uint16_t uuid
  特征的UUID
- uint16_t hdl
  handle
- uint8_t sz
  写数据长度
- uint8_t data [MAX_ATT_DATA_SZ]
  写数据

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

详细描述

att写事件

结构体成员变量说明

uint16_t gap_att_write_evt::primary

写handle的主要服务

uint16_t gap_att_write_evt::uuid

特征的UUID

uint16_t gap_att_write_evt::hdl

handle

uint8_t gap_att_write_evt::sz

写数据长度

uint8_t gap_att_write_evt::data[MAX_ATT_DATA_SZ]

写数据

---

gap_ble_addr结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint8_t type
  蓝牙地址类型，参考 BLE_ADDRESS_TYPE
- uint8_t addr [BD_ADDR_SZ]
  蓝牙地址

详细描述

蓝牙地址类型和蓝牙地址

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

芯片微 Confidential

结构体成员变量说明

uint8_t gap_ble_addr::type

蓝牙地址类型，参考_BLE_ADDRESS_TYPE_

uint8_t gap_ble_addr::addr[BD_ADDR_SZ]

蓝牙地址

---

gap_ble_evt结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint8_t evt_type
  *事件类型 EVT_TYPE_*
- uint32_t evt_code
  *事件码，参考 GAP_EVT_*
- union {
- struct gap_disconnected_evt disconn_evt
  GAP_EVT_DISCONNECTED
- struct gap_ble_addr bond_dev_evt
  GAP_EVT_CONNECTED
- struct gap_pairing_comp_evt pairing_comp_evt
  GAP_EVT_PAIRING_COMP
- struct gap_att_read_evt att_read_evt
  GAP_EVT_ATT_READ
- struct gap_att_write_evt att_write_evt
  GAP_EVT_ATT_WRITE
- struct gap_att_pre_write_evt att_pre_write_evt
  GAP_EVT_ATT_PREPARE_WRITE
- struct gap_att_exec_write_evt att_exec_write_evt
  GAP_EVT_ATT_EXECUTE_WRITE
- struct gap_att_handle_configure_evt att_handle_config_evt
  GAP_EVT_ATT_HANDLE_CONFIGURE
- struct gap_l2cap_update_rsp_evt l2cap_update_rsp_evt
  GAP_EVT_L2CAP_UPDATE_RSP
- struct gap_link_params conn_update_complete_evt
  GAP_EVT_CONN_UPDATE_COMP
- struct gap_scan_report_evt scan_report_evt
  GAP_EVT_SCAN_REPORT
- struct gap_scan_end_evt scan_end_evt
  GAP_EVT_SCAN_END
- } evt

成讯微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

详细描述

GAP所有事件

结构体成员变量说明

uint8_t gap_ble_evt::evt_type

事件类型 EVT_TYPE

uint32_t gap_ble_evt::evt_code

事件码，参考 GAP_EVT_

struct gap_disconnected_evt gap_ble_evt::disconn_evt

GAP_EVT_DISCONNECTED

struct gap_ble_addr gap_ble_evt::bond_dev_evt

GAP_EVT_CONNECTED

struct gap_pairing_comp_evt gap_ble_evt::pairing_comp_evt

GAP_EVT_PAIRING_COMP

struct gap_att_read_evt gap_ble_evt::att_read_evt

GAP_EVT_ATT_READ

struct gap_att_write_evt gap_ble_evt::att_write_evt

GAP_EVT_ATT_WRITE

struct gap_att_pre_write_evt gap_ble_evt::att_pre_write_evt

GAP_EVT_ATT_PREPARE_WRITE

struct gap_att_exec_write_evt gap_ble_evt::att_exec_write_evt

GAP_EVT_ATT_EXECUTE_WRITE

struct gap_att_handle_configure_evt gap_ble_evt::att_handle_config_evt

GAP_EVT_ATT_HANDLE_CONFIGURE

芯片微 Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

struct gap_l2cap_update_rsp_evt gap_ble_evt::l2cap_update_rsp_evt

GAP_EVT_L2CAP_UPDATE_RSP

struct gap_link_params gap_ble_evt::conn_update_complete_evt

GAP_EVT_CONN_UPDATE_COMP

struct gap_scan_report_evt gap_ble_evt::scan_report_evt

GAP_EVT_SCAN_REPORT

struct gap_scan_end_evt gap_ble_evt::scan_end_evt

GAP_EVT_SCAN_END

union { ... } gap_ble_evt::evt

---

gap_bond_dev结构体 参考

```
#include <lib_cn.h>
```

成员变量

- struct gap_ble_addr addr
- struct gap_key_params key

详细描述

配对绑定的参数

结构体成员变量说明

struct gap_ble_addr gap_bond_dev::addr

struct gap_key_params gap_bond_dev::key

---

gap_disconnected_evt结构体 参考

```
#include <lib_cn.h>
```


All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

SYDTEK Confidential

成员变量

- uint8_t reason
  断连原因，参考蓝牙Spec的HCI error codes

详细描述

断连事件

结构体成员变量说明

uint8_t gap_disconnected_evt::reason

断连原因，参考蓝牙Spec的HCI error codes

---

gap_evt_callback结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint32_t evt_mask
  *上报事件mask，对应bit置1则不上报该事件，bitmap参考 GAP_EVT_*
- void(* p_callback)(struct gap_ble_evt *p_evt)
  事件回调函数

详细描述

GAP事件回调，通过该结构注册回调函数，同时控制上报哪些事件

结构体成员变量说明

uint32_t gap_evt_callback::evt_mask

上报事件mask，对应bit置1则不上报该事件，bitmap参考 GAP_EVT_

void(* gap_evt_callback::p_callback) (struct gap_ble_evt *p_evt)

事件回调函数

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

gap_key_params结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint8_t ediv [MAX_EDIV_SZ]
- uint8_t rand [MAX_RAND_SZ]
- uint8_t ltk [MAX_KEY_SZ]
- uint8_t iden [MAX_KEY_SZ]

详细描述

配对绑定的key

结构体成员变量说明

uint8_t gap_key_params::ediv[MAX_EDIV_SZ]

uint8_t gap_key_params::rand[MAX_RAND_SZ]

uint8_t gap_key_params::ltk[MAX_KEY_SZ]

uint8_t gap_key_params::iden[MAX_KEY_SZ]

---

gap_l2cap_update_rsp_evt结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint16_t result
  更新回应结果，参考L2CAP_UPDATE_RSP_RES

详细描述

连接参数更新回应事件

结构体成员变量说明

uint16_t gap_l2cap_update_rsp_evt::result

更新回应结果，参考L2CAP_UPDATE_RSP_RES

盛士微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

gap_link_params结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint16_t interval
- uint16_t latency
- uint16_t svto

详细描述

链路更新参数

结构体成员变量说明

uint16_t gap_link_params::interval

uint16_t gap_link_params::latency

uint16_t gap_link_params::svto

gap_pairing_comp_evt结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint8_t status
  配对完成状态，参考GAP_RET_STATUS
- uint8_t dir
  配对失败发起方，参考DIR_IN 和DIR_OUT，如果配对成功则忽略该字段
- uint8_t reason
  配对失败原因，参考SMP_FAILED_CODE，如果配对成功则忽略该字段
- struct gap_key_params enc_key
  加密key参数

详细描述

配对完成事件

芯微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

结构体成员变量说明

uint8_t gap_pairing_comp_evt::status

配对完成状态，参考GAP_RET_STATUS

uint8_t gap_pairing_comp_evt::dir

配对失败发起方，参考DIR_IN 和DIR_OUT，如果配对成功则忽略该字段

uint8_t gap_pairing_comp_evt::reason

配对失败原因，参考SMP_FAILED_CODE，如果配对成功则忽略该字段

struct gap_key_params gap_pairing_comp_evt::enc_key

加密key参数

---

gap_scan_dev结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint8_t adv_type
  广播类型，参考 ADV_CH_PKT_TYPE
- struct gap_ble_addr peer_addr

详细描述

扫描到对端设备的广播

结构体成员变量说明

uint8_t gap_scan_dev::adv_type

广播类型，参考 ADV_CH_PKT_TYPE

struct gap_ble_addr gap_scan_dev::peer_addr

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

芯讯通 Confidential

gap_scan_end_evt结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint8_t dev_cnt
  扫描到的设备数
- struct gap_scan_dev * p_dev_list
  扫描到的设备地址列表

详细描述

扫描周期结束事件

结构体成员变量说明

uint8_t gap_scan_end_evt::dev_cnt

扫描到的设备数

struct gap_scan_dev* gap_scan_end_evt::p_dev_list

扫描到的设备地址列表

---

gap_scan_params结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint8_t mode
  扫描模式，参考SCAN_TYPE
- uint8_t channel
  扫描通道bitmap，对应bit设置1为使能，bit0: ch37, bit1: ch38, bit2: ch39
- uint16_t interval
  扫描间隔，单位：0.625ms，取值范围0x0020~0x4000，即(20ms to 10.24s)
- uint16_t window
  扫描窗口，单位：0.625ms，取值范围0x0020~0x4000，即(20ms to 10.24s)，取值应当小于interval
- uint16_t timeout
  扫描周期，单位：1s，取值范围0x0001~0x3FFF，当设置为0x0000时只做单次扫描

芯片微 Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

详细描述

扫描参数设置，在开始扫描之前必须要设置该参数

结构体成员变量说明

uint8_t gap_scan_params::mode

扫描模式，参考SCAN_TYPE

uint8_t gap_scan_params::channel

扫描通道bitmap，对应bit设置1为使能，bit0: ch37, bit1: ch38, bit2: ch39

uint16_t gap_scan_params::interval

扫描间隔，单位：0.625ms，取值范围0x0020~0x4000，即(20ms to 10.24s)

uint16_t gap_scan_params::window

扫描窗口，单位：0.625ms，取值范围0x0020~0x4000，即(20ms to 10.24s)，取值应当小于interval

uint16_t gap_scan_params::timeout

扫描周期，单位：1s，取值范围0x0001~0x3FFF，当设置为0x0000时只做单次扫描

---

gap_scan_report_evt结构体 参考

```
#include <lib_cn.h>
```

成员变量

- struct gap_scan_dev dev
  扫描到的设备
- uint8_t data_len
  广播数据长度
- uint8_t adv_data [MAX_ADV_DATA_SZ]
  广播数据
- uint8_t rssi
  接收信号强度指示

详细描述

扫描数据上报事件，当扫描到一个广播包时，会通过该事件上报扫描的广播数据

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

芯片微Confidential

结构体成员变量说明

struct gap_scan_dev gap_scan_report_evt::dev

扫描到的设备

uint8_t gap_scan_report_evt::data_len

广播数据长度

uint8_t gap_scan_report_evt::adv_data[MAX_ADV_DATA_SZ]

广播数据

uint8_t gap_scan_report_evt::rssi

接收信号强度指示

---

gap_update_params结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint16_t updateitv_min
  最小连接间隔，单位：1.25ms，取值范围：0x0006~0x0c80，即7.5ms~4s
- uint16_t updateitv_max
  最大连接间隔，单位：1.25ms，取值范围：0x0006~0x0c80，即7.5ms~4s
- uint16_t updatelatency
  连接latency，
- uint16_t updatesvto

详细描述

连接参数更新请求参数

结构体成员变量说明

uint16_t gap_update_params::updateitv_min

最小连接间隔，单位：1.25ms，取值范围：0x0006~0x0c80，即7.5ms~4s

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

uint16_t gap_update_params::updateitv_max

最大连接间隔，单位：1.25ms，取值范围：0x0006~0x0c80，即7.5ms~4s

uint16_t gap_update_params::updatelatency

连接latency，

uint16_t gap_update_params::updatesvto

---

gap_wakeup_config结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint8_t wakeup_type
  WAKEUP_TYPE
- bool timer_wakeup_en
  使能timer中断唤醒
- bool gpi_wakeup_en
  使能gpio中断唤醒
- bool wdt_wakeup_en
  使能watchdog中断唤醒
- bool rtc_wakeup_en
  使能RTC 中断唤醒
- bool capdet_wakeup_en
  使能capdet中断唤醒
- bool ana_wakeup_en
  使能analog中断唤醒
- uint32_t gpi_wakeup_cfg
  gpio 唤醒源配置，bit0: gpio0 ...，当使能gpi_wakeup_en时生效

详细描述

唤醒配置参数，对应系统sleep模式的唤醒配置，sleep模式包括 mcu sleep和powerdown 2种，2种模式下均可通过如下配置的唤醒源唤醒， 为方便应用使用，系统提供丰富的唤醒源且可同时设置

结构体成员变量说明

uint8_t gap_wakeup_config::wakeup_type

WAKEUP_TYPE

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

盛达微 Confidential

bool gap_wakeup_config::timer_wakeup_en

使能timer中断唤醒

bool gap_wakeup_config::gpi_wakeup_en

使能gpio中断唤醒

bool gap_wakeup_config::wdt_wakeup_en

使能watchdog中断唤醒

bool gap_wakeup_config::rtc_wakeup_en

使能RTC 中断唤醒

bool gap_wakeup_config::capdet_wakeup_en

使能capdet中断唤醒

bool gap_wakeup_config::ana_wakeup_en

使能analog中断唤醒

uint32_t gap_wakeup_config::gpi_wakeup_cfg

gpio 唤醒源配置，bit0: gpio0 ...，当使能gpi_wakeup_en时生效

---

smp_pairing_req结构体 参考

```
#include <lib_cn.h>
```

成员变量

- uint8_t io
  SMP_IO_CAPABILITY
- uint8_t oob
  SMP_OOB_FLAG
- uint8_t flags:2
  SMP_BONDING_FLAGS
- uint8_t mitm:1
  0：不请求MITM保护，1：请求MITM保护
- uint8_t rsvd:5
- uint8_t max_enc_sz
  min: MIN_KEY_SZ , max: MAX_KEY_SZ

盛士微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

- uint8_t init_key
  SMP_KEY_DISTRIBUTION
- uint8_t rsp_key
  SMP_KEY_DISTRIBUTION

详细描述

配对请求参数设置，参考蓝牙Spec

结构体成员变量说明

uint8_t smp_pairing_req::io

SMP_IO_CAPABILITY

uint8_t smp_pairing_req::oob

SMP_OOB_FLAG

uint8_t smp_pairing_req::flags

SMP_BONDING_FLAGS

uint8_t smp_pairing_req::mitm

0：不请求MITM保护，1：请求MITM保护

uint8_t smp_pairing_req::rsvd

uint8_t smp_pairing_req::max_enc_sz

min: MIN_KEY_SZ , max: MAX_KEY_SZ

uint8_t smp_pairing_req::init_key

SMP_KEY_DISTRIBUTION

uint8_t smp_pairing_req::rsp_key

SMP_KEY_DISTRIBUTION

盛大微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

文件说明

Include/lib_cn.h 文件参考

该文件为SYD_8811低功耗蓝牙芯片开发头文件，

使用该芯片完成蓝牙开发的程序必须包含该头文件，

文件中包含所有蓝牙开发的宏定义，结构体定义，以及相关的接口定义.

结构体

- struct gap_ble_addr
- struct gap_key_params
- struct gap_bond_dev
- struct gap_adv_params
- struct gap_scan_params
- struct gap_scan_dev
- struct gap_scan_report_evt
- struct gap_scan_end_evt
- struct gap_update_params
- struct gap_disconnected_evt
- struct gap_pairing_comp_evt
- struct gap_att_read_evt
- struct gap_att_write_evt
- struct gap_att_pre_write_evt
- struct gap_att_exec_write_evt
- struct gap_att_handle_configure_evt
- struct gap_l2cap_update_rsp_evt
- struct gap_link_params
- struct gap_ble_evt
- struct gap_evt_callback
- struct gap_att_report
- struct gap_att_report_handle
- struct gap_wakeup_config
- struct smp_pairing_req
- struct att_err_rsp
- struct att_mtu_rsp
- struct att_find_info_16
- struct att_find_info_128
- union att_find_info_payload
- struct att_find_info_rsp
- struct att_find_by_type_val_rsp
- struct att_read_by_type_service_16
- struct att_read_by_type_service_128
- union att_read_by_type_service_payload
- struct att_read_by_type_service_rsp
- struct att_read_by_type_16
- struct att_read_by_type_128
- union att_read_by_type_payload
- struct att_read_by_type_rsp
- struct att_read_by_type_pair_val
- struct att_read_by_type_val_rsp
- struct att_read_by_type_chartextend_rsp
- struct att_read_by_type_include_rsp
- struct att_read_rsp
- struct att_read_blob_rsp
- struct att_read_multiple_rsp
- struct att_read_by_group_type_16

SYDTEK Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

- struct att_read_by_group_type_128
- union att_read_by_group_type_payload
- struct att_read_by_group_type_rsp
- struct att_hdl_val_notifivation
- struct att_hdl_val_indication
- struct attc_ble_evt

宏定义

- #define true 1
- #define false 0
- #define MAX_ATT_REPORT_HDL 20
  最大ATT通知和指示handle数
- #define MAX_ATT_DATA_SZ 23
  最大ATT数据长度
- #define BD_ADDR_SZ 6
  蓝牙地址长度
- #define MAX_KEY_SZ 16
  最大加密key长度
- #define MIN_KEY_SZ 7
  最小加密key长度
- #define MAX_RAND_SZ 8
  RAND长度
- #define MAX_EDIV_SZ 2
  EDIV长度
- #define ACCESS_CODE_SZ 4
  蓝牙接入码长度
- #define MAX_ADV_DATA_SZ 31
  最大广播数据长度
- #define MAX_SCAN_DEV_NUM 8
  最大扫描设备个数
- #define DIR_IN 0
  方向为从远端设备接收
- #define DIR_OUT 1
  方向为从本地设备发送

类型定义

- typedef unsigned char bool
- typedef void(* sys_timer_cb) (void)
- typedef void(* p_attc_callback) (struct attc_ble_evt *p_evt)

枚举

- enum _RETURN_STATUS_ { _PARAM_ERROR_ = 0x00, _NO_ERROR_ = 0x01 }
- enum _COMPANY_ID_ { COMPANY_ID_SYD = 0x0 }
- enum _QFN_TYPE_ { T_QFN_48 = 0, T_QFN_32 = 1 }
- enum SCAN_TYPE { PASSIVE_SCAN = 0x00, ACTIVE_SCAN = 0x01 }
- enum _BLE_ADDRESS_TYPE_ { PUBLIC_ADDRESS_TYPE = 0x00, RANDOM_ADDRESS_TYPE = 0x01 }
- enum GAP_RET_STATUS { RET_FAIL = 0x00, RET_SUCCESS = 0x01 }
- enum _EVT_TYPE_ { GAP_EVT = 0x00 }
- enum _GAP_EVT_ { GAP_EVT_ADV_END = 0x00000001, GAP_EVT_CONNECTED = 0x00000002, GAP_EVT_DISCONNECTED = 0x00000004, GAP_EVT_PAIRING_COMP = 0x00000008, GAP_EVT_PASSKEY_REQ = 0x00000010, GAP_EVT_SHOW_PASSKEY_REQ

芯讯微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

= 0x00000020, GAP_EVT_CONNECTION_INTERVAL = 0x00000040, GAP_EVT_CONNECTION_SLEEP = 0x00000080, GAP_EVT_ATT_READ = 0x00000100, GAP_EVT_ATT_WRITE = 0x00000200, GAP_EVT_ATT_PREPARE_WRITE = 0x00000400, GAP_EVT_ATT_EXECUTE_WRITE = 0x00000800, GAP_EVT_ATT_HANDLE_CONFIRMATION = 0x00001000, GAP_EVT_ATT_HANDLE_CONFIGURE = 0x00002000, GAP_EVT_ENC_START = 0x00004000, GAP_EVT_L2CAP_UPDATE_RSP = 0x00008000, GAP_EVT_CONN_UPDATE_COMP = 0x00010000, GAP_EVT_SCAN_REPORT = 0x00020000, GAP_EVT_SCAN_END = 0x00040000, GAP_EVT_PAIRING_START = 0x00080000 }

- enum _ADV_CH_PKT_TYPE_ { ADV_IND = 0x00, ADV_DIRECT_IND = 0x01, ADV_NOCONN_IND = 0x02, SCAN_REQ = 0x03, SCAN_RSP = 0x04, CONNECT_REQ = 0x05, ADV_SCAN_IND = 0x06 }
- enum BLE_SEND_TYPE { BLE_GATT_NOTIFICATION = 0x0001, BLE_GATT_INDICATION = 0x0002 }
- enum WAKEUP_TYPE { SLEEP_WAKEUP = 0, POWERDOWN_WAKEUP = 1 }
- enum SMP_IO_CAPABILITY { IO_DISPLAY_ONLY = 0x00, IO_DISPLAY_YESNO = 0x01, IO_KEYBOARD_ONLY = 0x02, IO_NO_INPUT_OUTPUT = 0x03, IO_KEYBOARD_DISPLAY = 0x04 }
- enum SMP_OOB_FLAG { OOB_AUTH_NOT_PRESENT = 0x00, OOB_AUTH_PRESENT = 0x01 }
- enum SMP_BONDING_FLAGS { AUTHREQ_NO_BONDING = 0x00, AUTHREQ_BONDING = 0x01 }
- enum SMP_KEY_DISTRIBUTION { SMP_KEY_MASTER_IDEN = 0x01, SMP_KEY_ADDR_INFO = 0x02, SMP_KEY_SIGNIN_INFO = 0x04 }
- enum SMP_FAILED_CODE { SMP_RESERVED = 0x00, SMP_PASSKEY_ENTRY_FAILED = 0x01, SMP_OOB_NOT_AVAILABLE = 0x02, SMP_AUTH_REQUIREMENTS = 0x03, SMP_CONFIRM_VALUE_FAILED = 0x04, SMP_PAIRING_NOT_SUPPORTED = 0x05, SMP_ENCTYPTION_KEY_SZ = 0x06, SMP_COMMAND_NOT_SUPPORTED = 0x07, SMP_UNSPECIFIED_REASON = 0x08, SMP_REPEATED_ATTEMPTS = 0x09, SMP_INVALID_PARAMETERS = 0x0A, SMP_FAIL_TIMEOUT = 0xFF }
- enum _SYSTEM_CLOCK_SEL_ { SYSTEM_CLOCK_64M_RCOSC = 0x00, SYSTEM_CLOCK_32M_RCOSC = 0x01, SYSTEM_CLOCK_16M_RCOSC = 0x02, SYSTEM_CLOCK_8M_RCOSC = 0x03, SYSTEM_CLOCK_4M_RCOSC = 0x04, SYSTEM_CLOCK_2M_RCOSC = 0x05, SYSTEM_CLOCK_1M_RCOSC = 0x06, SYSTEM_CLOCK_500k_RCOSC = 0x07, SYSTEM_CLOCK_32M_XOSC = 0x08 }
- enum _32K_CLOCK_SEL_ { SYSTEM_32K_CLOCK_RCOSC = 0x00, SYSTEM_32K_CLOCK_XOSC = 0x01, SYSTEM_32K_CLOCK_32MXO_DIV = 0x02 }
- enum SOFT_TIMER_CTRL_TYPE { SOFT_TIMER_0 = 0x0020, SOFT_TIMER_1 = 0x0040, SOFT_TIMER_2 = 0x0080, SOFT_TIMER_3 = 0x0100, SOFT_TIMER_4 = 0x0200 }
- enum L2CAP_UPDATE_RSP_RES { CONN_PARAMS_ACCEPTED = 0x0000, CONN_PARAMS_REJECTED = 0x0001 }
- enum ATT_CMD_CODE { ATT_ERR_RSP = 0x01, ATT_MTU_REQ = 0x02, ATT_MTU_RSP = 0x03, ATT_FIND_INFO_REQ = 0x04, ATT_FIND_INFO_RSP = 0x05, ATT_FIND_BY_TYPE_VALUE_REQ = 0x06, ATT_FIND_BY_TYPE_VALUE_RSP = 0x07, ATT_READ_BY_TYPE_REQ = 0x08, ATT_READ_BY_TYPE_RSP = 0x09, ATT_READ_REQ = 0x0A, ATT_READ_RSP = 0x0B, ATT_READ_BLOB_REQ = 0x0C, ATT_READ_BLOB_RSP = 0x0D, ATT_READ_MULTIPLE_REQ = 0x0E, ATT_READ_MULTIPLE_RSP = 0x0F, ATT_READ_BY_GROUP_TYPE_REQ = 0x10, ATT_READ_BY_GROUP_TYPE_RSP = 0x11, ATT_WRITE_REQ = 0x12, ATT_WRITE_RSP = 0x13, ATT_WRITE_CMD = 0x52, ATT_PREPARE_WRITE_REQ = 0x16, ATT_PREPARE_WRITE_RSP = 0x17, ATT_EXECUTE_WRITE_REQ = 0x18, ATT_EXECUTE_WRITE_RSP = 0x19, ATT_HANDLE_VAL_NOTIFICATION = 0x1B, ATT_HANDLE_VAL_INDICATION = 0x1D, ATT_HANDLE_VAL_CONFIRMATION = 0x1E, ATT_SIGNED_WRITE_CMD = 0xD2 }
- enum ATT_ERROR_CODE { ATT_INVALID_HANDLE = 0x01, ATT_READ_NOT_PEMITTED = 0x02, ATT_WRITE_NOT_PEMITTED = 0x03, ATT_INVALID_PDU = 0x04, ATT_INSUFFICIENT_AUTHEN = 0x05, ATT_REQ_NOT_SUPPORTED = 0x06, ATT_INVALID_OFFSET = 0x07, ATT_INSUFFICIENT_AUTHOR = 0x08, ATT_PREPARE_QUEUE_FULL = 0x09,

Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

ATT_ATTRIBUTE_NOT_FOUND = 0x0A, ATT_ATTRIBUTE_NOT_LONG = 0x0B, ATT_INSUFFICIENT_ENC_KEY_SZ = 0x0C, ATT_INVALID_ATTRIBUTE_VAL_LEN = 0x0D, ATT_UNLIKELY_ERROR = 0x0E, ATT_INSUFFICIENT_ENC = 0x0F, ATT_UNSUPPORTED_GROUP_TYPE = 0x10, ATT_INSUFFICIENT_RESOURCES = 0x11 }

- enum ATT_CHAR_PROPERTY { ATT_CHAR_BROADCAST = 0x01, ATT_CHAR_READ = 0x02, ATT_CHAR_WEIRE_WO_RSP = 0x04, ATT_CHAR_WEIRE = 0x08, ATT_CHAR_NOTIFY = 0x10, ATT_CHAR_INDICATE = 0x20, ATT_CHAR_AUTH_SIGNED_WRITE = 0x40, ATT_CHAR_EXTEND_PROPERTY = 0x80 }
- enum _GATT_FIND_INFO_UUID_TYPE_ { GATT_FIND_INFO_UUID_16 = 0x01, GATT_FIND_INFO_UUID_128 = 0x02 }
- enum ATT_EXEC_WRITE_FLAGS { ATT_EXEC_WRITE_CANCEL = 0, ATT_EXEC_WRITE_IMMED = 1 }

函数

- uint8_t BleInit (void)
  低功耗蓝牙初始化函数
- uint8_t DisConnect (void)
  主动断开蓝牙连接，
- uint8_t SetDevAddr (struct gap_ble_addr *p_dev_addr)
  设置蓝牙地址
- uint8_t GetDevAddr (struct gap_ble_addr *p_dev_addr)
  获取当前蓝牙地址
- uint8_t SetLEFeature (uint8_t *p_feature)
  设置蓝牙特性支持
- uint8_t SetAdvParams (struct gap_adv_params *p_adv_params)
  设置广播参数
- uint8_t SetAdvData (uint8_t *p_adv, uint8_t adv_sz, uint8_t *p_scan, uint8_t sacn_sz)
  设置广播数据和扫描回应数据
- uint8_t StartAdv (void)
  开始广播

  广播类型，广播间隔，广播周期等设置见接口SetAdvParams

  广播数据设置见接口SetAdvData
- uint8_t StopAdv (void)
  停止广播
- uint8_t SetScanParams (struct gap_scan_params *p_scan_params)
  扫描参数设置
- uint8_t StartScan (void)
  开始扫描
- uint8_t StopScan (void)
  停止扫描
- uint8_t SetSecParams (struct smp_pairing_req *p_sec_params)
  设置配对请求参数

  slave角色会在配对回应协议中发送， master角色会在配对请求协议中发送
- uint8_t SetPasskey (uint32_t passkey)
  设置passkey，在配对过程中根据连接双方的io能力确定

  是否需要输入或显示一个passkey，如果是justwork则无需设置
- uint8_t SecurityReq (uint8_t flag, uint8_t mitm)
  当slave主动要求发起配对时，调用该接口
- uint8_t SetConnectionUpdate (struct gap_update_params *p_update_params)

Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

当slave主动要求更新连接参数时，调用该接口

- uint8_t GetLinkParameters (struct gap_link_params *p_link)
  获取当前链路的连接参数
- uint8_t SetWinWideMinusCnt (uint8_t cnt)
- uint8_t ConnectionLatencyMode (uint8_t en)
  连接latency模式控制，latency模式请参考蓝牙Spec
- uint8_t SetEvtCallback (struct gap_evt_callback *p_callback)
  设置GAP事件回调函数，当需要接收任何协议栈事件通知时， 都需要调用该接口来注册回调函数

  在协议栈事件发生时，会调用注册的 回调函数来通知对应的事件

  以事件回调方式来完成协议栈到应用程序的通信， 反过来应用程序到协议栈的通信方式为接口调用
- uint8_t GetGATTReportHandle (struct gap_att_report_handle **p_hdl)
  获取GATT所有的通知和指示信息列表
- uint8_t SetGATTReadRsp (uint8_t len, uint8_t *p_data)
  设置GATT需要读取的数据
- uint8_t CheckFIFOFull (void)
  检查协议栈FIFO是否已满
- uint8_t GATTDataSend (uint8_t type, struct gap_att_report *p_report, uint8_t len, uint8_t *p_data)
  gatt数据发送
- uint8_t Rand (void)
  随机数生成
- void DelayUS (uint16_t dly)
  延迟函数
- void DelayMS (uint32_t dly)
  延迟函数
- uint8_t GetCompanyID (void)
  获取公司ID
- uint8_t GetQFNType (void)
  获取芯片QFN 封装类型
- void RFRead (uint8_t addr, uint8_t *data)
- void RFWrite (uint8_t addr, uint8_t data)
- void ble_sched_execute (void)
  协议栈日程执行
- bool ble_sched_finish (void)
  检查协议栈日程是否执行完成
- uint8_t LPOCalibration (void)
  内部32k LPO时钟校准
- uint8_t RCOSCCalibration (void)
  内部RCOSC时钟校准
- uint8_t MCUClockSwitch (uint8_t sel)
  mcu时钟切换
- uint8_t ClockSwitch (uint8_t sel)
  32k 时钟切换
- uint8_t GetMCUClock (uint8_t *p_sel)
  获取当前mcu时钟
- uint8_t GetClock (uint8_t *p_sel)

芯微 Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

获取当前32k 时钟

- uint8_t TimerStart (uint16_t type, uint32_t timecnt_100ms, bool reload, sys_timer_cb pfnCallback)
  启动一个内部软定时器
- void TimerStop (uint16_t type)
  停止一个内部软定时器
- uint8_t WakeupConfig (struct gap_wakeup_config *p_cfg)
  系统唤醒配置接口
- uint8_t LLSleep (void)
- uint8_t SystemSleep (void)
  MCU进入sleep状态
- uint8_t SystemPowerDown (void)
  系统下电
- uint8_t SystemReset (void)
  系统软复位
- uint8_t RFSleep (void)
- uint8_t RFWakeup (void)
- uint8_t UartEn (uint8_t en)
  在RF sleep时，配置uart使能工作
- uint8_t SetBondManagerIndex (uint8_t idx)
  设置当前bonding的索引位置
- uint8_t GetBondDevice (struct gap_bond_dev *p_device)
  获取当前bonding设备的信息
- uint8_t AddBondDevice (struct gap_bond_dev *p_device)
  增加bonding设备信息
- uint8_t DelBondDevice (void)
  删除bonding设备信息
- uint8_t DelAllBondDevice (void)
  删除所有bonding设备信息
- uint8_t ReadProfileData (uint16_t addr, uint16_t len, uint8_t *p_buf)
  读profile数据
- uint8_t WriteProfileData (uint16_t addr, uint16_t len, uint8_t *p_buf)
  写profile数据
- uint8_t EraseFlashData (uint32_t addr, uint8_t sector_num)
  擦出flash数据
- uint8_t ReadFlashData (uint32_t addr, uint16_t len, uint8_t *p_buf)
  读flash数据
- uint8_t WriteFlashData (uint32_t addr, uint16_t len, uint8_t *p_buf)
  写flash数据
- uint8_t CodeErase (void)
  擦出Firmware code
- uint8_t CodeWrite (uint16_t offset, uint16_t len, uint8_t *p_buf)
  写Firmware code
- uint8_t CodeUpdate (uint8_t *p_desc, uint8_t *p_ver, uint16_t sz, uint16_t checksum)
  更新Firmware code 描述，检查checkum
- uint8_t ATTCSetCallback (p_attc_callback pfn_callback)
  ATT Client设置回调函数接口
- uint8_t ATTCMTUReq (uint16_t mtu)
  ATT Client MTU Request

Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

- uint8_t ATTCFindInfoReq (uint16_t start_hdl, uint16_t end_hdl)
  ATT Client Find Information Request
- uint8_t ATTCFindByTypeValueReq (uint16_t start_hdl, uint16_t end_hdl, uint16_t type, uint8_t val_sz, uint8_t *p_val)
  ATT Client Find By Type Value Request
- uint8_t ATTCReadByTypeReq (uint16_t start_hdl, uint16_t end_hdl, uint16_t type_sz, uint8_t *p_type)
  ATT Client Read By Type Request
- uint8_t ATTCReadReq (uint16_t hdl)
  ATT Client Read Request
- uint8_t ATTCReadBlobReq (uint16_t hdl, uint16_t offset)
  ATT Client Read Blob Request
- uint8_t ATTCReadMultipleReq (uint8_t hdl_sz, uint8_t *p_hdl)
  ATT Client Read Multiple Request
- uint8_t ATTCReadByGroupTypeReq (uint16_t start_hdl, uint16_t end_hdl, uint16_t type_sz, uint8_t *p_type)
  ATT Client Read By Group Type Request
- uint8_t ATTCWriteReq (uint16_t hdl, uint16_t sz, uint8_t *p_buf)
  ATT Client Write Request
- uint8_t ATTCWriteCmdReq (uint16_t hdl, uint16_t sz, uint8_t *p_buf)
  ATT Client Write Command Request
- uint8_t ATTCPrepareWriteReq (uint16_t hdl, uint16_t offset, uint16_t sz, uint8_t *p_buf)
  ATT Client Prepare Write Request
- uint8_t ATTCExecuteWriteReq (uint8_t flags)
  ATT Client Execute Write Request
- uint8_t ATTCConfirmation (void)
  ATT Client Confirmation

详细描述

该文件为SYD_8811低功耗蓝牙芯片开发头文件，

使用该芯片完成蓝牙开发的程序必须包含该头文件，

文件中包含所有蓝牙开发的宏定义，结构体定义，以及相关的接口定义.

作者:

版本:

日期:

2018

版权所有:

SYD Copyright.

芯片微 Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

宏定义说明

#define true 1

#define false 0

#define MAX_ATT_REPORT_HDL 20

最大ATT通知和指示handle数

#define MAX_ATT_DATA_SZ 23

最大ATT数据长度

#define BD_ADDR_SZ 6

蓝牙地址长度

#define MAX_KEY_SZ 16

最大加密key长度

#define MIN_KEY_SZ 7

最小加密key长度

#define MAX_RAND_SZ 8

RAND长度

#define MAX_EDIV_SZ 2

EDIV长度

#define ACCESS_CODE_SZ 4

蓝牙接入码长度

#define MAX_ADV_DATA_SZ 31

最大广播数据长度

#define MAX_SCAN_DEV_NUM 8

最大扫描设备个数

芯片微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

#define DIR_IN 0

方向为从远端设备接收

#define DIR_OUT 1

方向为从本地设备发送

类型定义说明

typedef unsigned char bool

typedef void(* sys_timer_cb) (void)

定时器timeout时的回调函数类型

typedef void(* p_attc_callback) (struct attc_ble_evt *p_evt)

Att客服端回调函数指针类型定义

枚举类型说明

enum _RETURN_STATUS_

接口调用返回状态

枚举值:

_PARAM_ERROR_	接口调用出错
_NO_ERROR_	接口调用成功

enum _COMPANY_ID_

公司ID

枚举值:

COMPANY_ID_SYD	

enum _QFN_TYPE_

QFN 类型

枚举值:

T_QFN_48	
T_QFN_32	

芯片微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

enum SCAN_TYPE

扫描类型

枚举值:

PASSIVE_SCAN	被动扫描，只扫描adv不回应
ACTIVE_SCAN	主动扫描，扫描到adv时会回应scan_req

enum _BLE_ADDRESS_TYPE_

蓝牙地址类型

枚举值:

PUBLIC_ADDRESS_TYPE	公有地址类型，定义参考蓝牙Spec.
RANDOM_ADDRESS_TYPE	随机地址类型，定义参考蓝牙Spec.

enum GAP_RET_STATUS

GAP返回状态

枚举值:

RET_FAIL	失败
RET_SUCCESS	成功

enum _EVT_TYPE_

协议栈事件类型

枚举值:

GAP_EVT	GAP事件

enum _GAP_EVT_

GAP事件类型

枚举值:

GAP_EVT_ADV_END	一个广播周期完成上报该事件，注：收到该事件表示广播停止
GAP_EVT_CONNECTED	连接建立上报该事件
GAP_EVT_DISCONNECTED	断开连接上报该事件

芯片微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

GAP_EVT_PAIRING_COMP	配到完成上报该事件
GAP_EVT_PASSKEY_REQ	配对过程中passkey输入请求事件
GAP_EVT_SHOW_PASSKEY_REQ	配对过程中passkey显示请求事件
GAP_EVT_CONNECTION_INTERVAL	每个连接间隔上报该事件，注：此事件上报太过频繁建议关闭
GAP_EVT_CONNECTION_SLEEP	每个连接完成后系统进入sleep上报该事件，注：此事件上报太过频繁建议关闭
GAP_EVT_ATT_READ	ATT读上报该事件
GAP_EVT_ATT_WRITE	ATT写上报该事件
GAP_EVT_ATT_PREPARE_WRITE	ATT预备写上报该事件
GAP_EVT_ATT_EXECUTE_WRITE	ATT执行写上报该事件
GAP_EVT_ATT_HANDLE_CONFIRMATION	ATT确认指示(indication)上报该事件
GAP_EVT_ATT_HANDLE_CONFIGURE	ATT客服端特征配置(Client Characteristic Configuration)上报事件
GAP_EVT_ENC_START	链路开始加密上报该事件
GAP_EVT_L2CAP_UPDATE_RSP	L2cap连接参数更新回应上报该事件，对应连接参数请求的回复
GAP_EVT_CONN_UPDATE_COMP	链路连接更新完成上报该事件
GAP_EVT_SCAN_REPORT	扫描设备上报该事件
GAP_EVT_SCAN_END	扫描完成上报该事件，完成条件包括 扫描设备数量达到最大或扫描周期完成，注：收到该事件表示扫描停止
GAP_EVT_PAIRING_START	配对开始上报该事件

enum _ADV_CH_PKT_TYPE_

广播通道协议包类型，定义参考蓝牙Spec

枚举值:

ADV_IND	

盛达微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

ADV_DIRECT_IND	
ADV_NOCONN_IND	
SCAN_REQ	
SCAN_RSP	
CONNECT_REQ	
ADV_SCAN_IND	

enum BLE_SEND_TYPE

Ble slave主动发送ATT数据类型

枚举值:

BLE_GATT_NOTIFICATION	通知
BLE_GATT_INDICATION	指示

enum WAKEUP_TYPE

系统唤醒类型

枚举值:

SLEEP_WAKEUP	mcu sleep唤醒
POWERDOWN_WAKEUP	系统下电(RTC有电)唤醒

enum SMP_IO_CAPABILITY

配对过程IO能力交换类型，定义参考蓝牙Spec

枚举值:

IO_DISPLAY_ONLY	
IO_DISPLAY_YESNO	
IO_KEYBOARD_ONLY	
IO_NO_INPUT_OUTPUT	
IO_KEYBOARD_DISPLAY	

enum SMP_OOB_FLAG

配对过程OOB表示标志，定义参考蓝牙Spec

枚举值:

OOB_AUTH_NOT_PRESENT	
OOB_AUTH_PRE	

芯微 Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

SENT	

enum SMP_BONDING_FLAGS

配对过程绑定标志，定义参考蓝牙Spec

枚举值:

AUTHREQ_NO_BONDING	
AUTHREQ_BONDING	

enum SMP_KEY_DISTRIBUTION

配对过程key分发bitmap，定义参考蓝牙Spec

枚举值:

SMP_KEY_MASTER_IDEN	
SMP_KEY_ADDR_INFO	
SMP_KEY_SIGNIN_INFO	

enum SMP_FAILED_CODE

配对过程失败码，定义参考蓝牙Spec

枚举值:

SMP_RESERVED	
SMP_PASSKEY_ENTRY_FAILED	
SMP_OOB_NOT_AVAILABLE	
SMP_AUTH_REQUIREMENTS	
SMP_CONFIRM_VALUE_FAILED	
SMP_PAIRING_NOT_SUPPORTED	
SMP_ENCTYPTION_KEY_SZ	
SMP_COMMAND_NOT_SUPPORTED	
SMP_UNSPECIFIED_REASON	
SMP_REPEATED_ATTEMPTS	
SMP_INVALID_PARAMETERS	
SMP_FAIL_TIMEOUT	

信微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

enum _SYSTEM_CLOCK_SEL_

系统时钟源类型

枚举值:

SYSTEM_CLOCK_64M_RCOSC	内部64M RCOSC
SYSTEM_CLOCK_32M_RCOSC	
SYSTEM_CLOCK_16M_RCOSC	
SYSTEM_CLOCK_8M_RCOSC	
SYSTEM_CLOCK_4M_RCOSC	
SYSTEM_CLOCK_2M_RCOSC	
SYSTEM_CLOCK_1M_RCOSC	
SYSTEM_CLOCK_500k_RCOSC	
SYSTEM_CLOCK_32M_XOSC	外部32M晶振

enum _32K_CLOCK_SEL_

系统RTC域32k时钟源类型

枚举值:

SYSTEM_32K_CLOCK_RCOSC	内部32k LPO
SYSTEM_32K_CLOCK_XOSC	外部32k晶振
SYSTEM_32K_CLOCK_32MXO_DIV	外部晶振分频

enum SOFT_TIMER_CTRL_TYPE

内部32k软定时器配置，共5个定时器可使用

枚举值:

SOFT_TIMER_0	
SOFT_TIMER_1	
SOFT_TIMER_2	
SOFT_TIMER_3	
SOFT_TIMER_4	

enum L2CAP_UPDATE_RSP_RES

连接参数更新回应结果

芯片微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

枚举值:

CONN_PARAMS_ACCEPTED	
CONN_PARAMS_REJECTED	

enum ATT_CMD_CODE

ATT PDU操作码，详细见Spec

枚举值:

ATT_ERR_RSP	
ATT_MTU_REQ	
ATT_MTU_RSP	
ATT_FIND_INFO_REQ	
ATT_FIND_INFO_RSP	
ATT_FIND_BY_TYPE_VALUE_REQ	
ATT_FIND_BY_TYPE_VALUE_RSP	
ATT_READ_BY_TYPE_REQ	
ATT_READ_BY_TYPE_RSP	
ATT_READ_REQ	
ATT_READ_RSP	
ATT_READ_BLOB_REQ	
ATT_READ_BLOB_RSP	
ATT_READ_MULTIPLE_REQ	
ATT_READ_MULTIPLE_RSP	
ATT_READ_BY_GROUP_TYPE_REQ	
ATT_READ_BY_GROUP_TYPE_RSP	
ATT_WRITE_REQ	
ATT_WRITE_RSP	
ATT_WRITE_CMD	
ATT_PREPARE_WRITE_REQ	
ATT_PREPARE_WRITE_RSP	
ATT_EXECUTE_WRITE_REQ	
ATT_EXECUTE_WRITE_RSP	

昆腾微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

ATT_HANDLE_VAL_NOTIFICATION	
ATT_HANDLE_VAL_INDICATION	
ATT_HANDLE_VAL_CONFIRMATION	
ATT_SIGNED_WRITE_CMD	

enum ATT_ERROR_CODE

ATT ATT_ERROR_RSP PDU错误码，详细见Spec

枚举值:

ATT_INVALID_HANDLE	
ATT_READ_NOT_PEMITTED	
ATT_WRITE_NOT_PEMITTED	
ATT_INVALID_PDU	
ATT_INSUFFICIENT_AUTHEN	
ATT_REQ_NOT_SUPPORTED	
ATT_INVALID_OFFSET	
ATT_INSUFFICIENT_AUTHOR	
ATT_PREPARE_QUEUE_FULL	
ATT_ATTRIBUTE_NOT_FOUND	
ATT_ATTRIBUTE_NOT_LONG	
ATT_INSUFFICIENT_ENC_KEY_SZ	
ATT_INVALID_ATTRIBUTE_VAL_LEN	
ATT_UNLIKELY_ERROR	
ATT_INSUFFICIENT_ENC	
ATT_UNSUPPORTED_GROUP_TYPE	
ATT_INSUFFICIENT_RESOURCES	

enum ATT_CHAR_PROPERTY

ATT特征申明的特征属性，详细见Spec

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

昇软微Confidential

枚举值:

ATT_CHAR_BROADCAST	
ATT_CHAR_READ	
ATT_CHAR_WEIRE_WO_RSP	
ATT_CHAR_WEIRE	
ATT_CHAR_NOTIFY	
ATT_CHAR_INDICATE	
ATT_CHAR_AUTH_SIGNED_WRITE	
ATT_CHAR_EXTEND_PROPERTY	

enum _GATT_FIND_INFO_UUID_TYPE_

Find Information UUID类型，16bit或128bit

枚举值:

GATT_FIND_INFO_UUID_16	
GATT_FIND_INFO_UUID_128	

enum ATT_EXEC_WRITE_FLAGS

ATT Execute Write Flags

枚举值:

ATT_EXEC_WRITE_CANCEL	取消所有的prepared writes
ATT_EXEC_WRITE_IMMED	立即写所有的prepared values

函数说明

uint8_t TimerStart (uint16_t *type*, uint32_t *timecnt_100ms*, bool *reload*, sys_timer_cb *pfnCallback*)

启动一个内部软定时器

注解:

该定时器采用100ms中断timer计时，当mcu sleep时每100ms都会唤醒mcu， 导致系统功耗增加，请谨慎使用

盛达微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com

参数:

type	- 要启动的定时器SOFT_TIMER_CTRL_TYPE
timecnt_100ms	- 定时器计数值，单位100ms
reload	- 是否重复启动
pfnCallback	- 定时器回调函数

返回:

时钟启动是否成功， RETURN_STATUS

void TimerStop (uint16_t *type*)

停止一个内部软定时器

参数:

type	- 要停止的定时器SOFT_TIMER_CTRL_TYPE

返回:

void

索引

INDEX

盛芯微Confidential

All rights reserved. Any portion in this paper shall not be reproduced, copied, or transformed to any other forms without permission. http://www.sydtek.com