libTSCANAPI.Interop 编程指导 V0.8

文档修订历史：

文件版本	日期	更新内容	备注
V1.00	2023.5.8	创建文档	

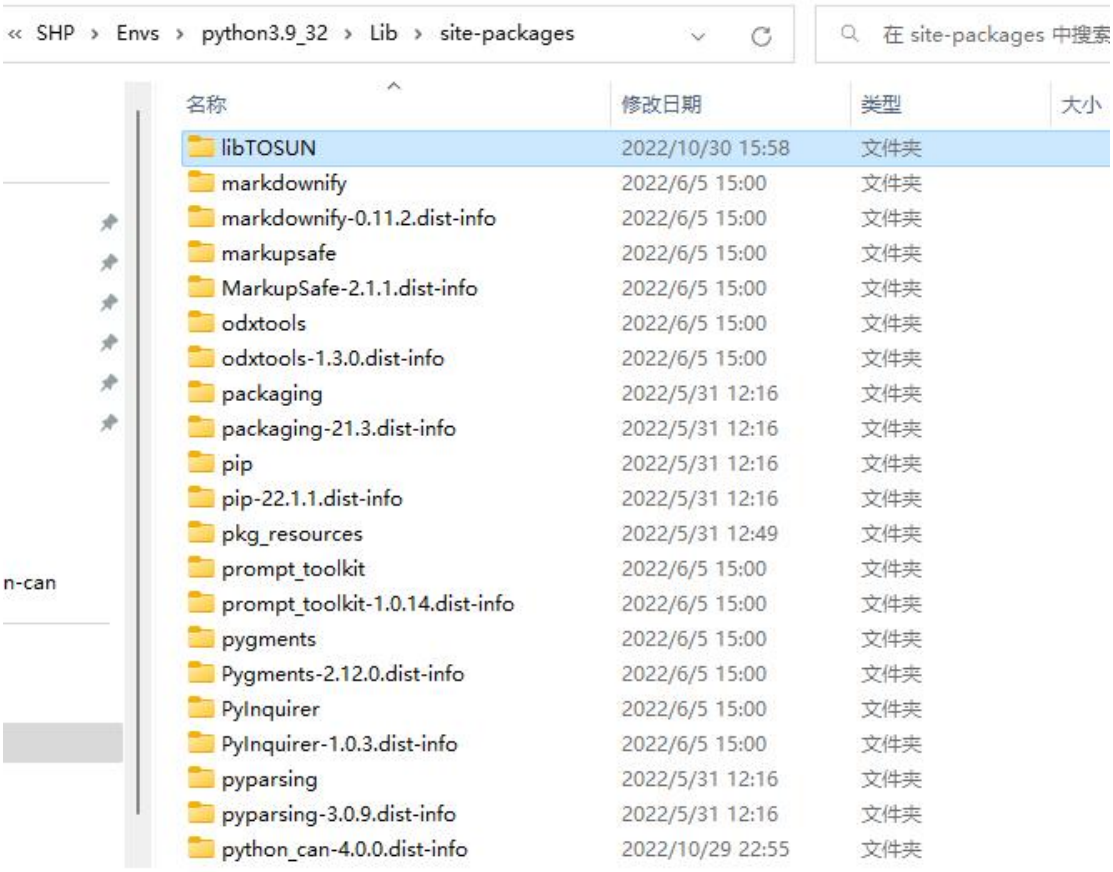
1. 目录

1. 什么情况下需要此文档？

用户基于 python 平台的编程语言，对上海同星智能科技有限公司的 TSCAN 系列工具（TSCANMini，TSLiteMini，TSCANFDMini，TSCANLINLite，TSCANLINPro）进行二次开发的时候，需要参考本文档，调用 API 函数来实现对设备的程序控制。

2. 添加库文件

1.将 libTOSUN 文件夹放置在 python 下的 site-packages 文件夹里如下所示：



2.在 python 工程文件中 from libTOSUN.libTOSUN import *即可使用本库。

3. 数据类型定义

1. TLIBCAN：CAN 总线数据类型

```
class TLIBCAN(Structure):
    _pack_ = 1
    _fields_ = [("FIdxChn", c_uint8),
                ("FProperties", c_uint8),
                ("FDLC", c_uint8),
                ("FReserved", c_uint8),
                ("FIdentifier", c_int32),
                ("FTimeUs", c_uint64),
                ("FData", c_uint8 * 8),
                ]

    def __init__(self, FIdxChn=0, FDLC=8, FIdentifier=0x1, FProperties=1, FData=[]):
        self.FIdxChn = FIdxChn
        self.FDLC = FDLC
        if self.FDLC > 8:
            self.FDLC = 8
        self.FIdentifier = FIdentifier
        self.FProperties = FProperties
        for i in range(len(FData)):
            self.FData[i] = FData[i]
```

成员：

FData：帧数据。最大长度为 8Bytes

FDLC：帧长度。

FIdentifier：帧 ID，如果为 0xFFFFFFFF，表示当前帧为错误帧

FIdxChn：帧通道，注意 CHANNEL_INDEX. CHN1 = 0, 实际上是从 0 开始计算的。

FTImeUS：帧时间戳，64 位 us 级时间戳。

FProperties：存储 CAN 相关的属性，比如是否远程帧，是否扩展帧。

其中，属性字节定义如下：

【1】 FProperties：CAN 属性定义：该参数默认为 0，共八个 bits，每一个位的定义如下：

Bit	意义
0	0：Rx 接收报文；1：Tx 发送报文
1	0：data frame 数据帧；1：remote frame 远程帧
2	0：std frame 标准帧；1：extended frame 扩展帧
3-5	Reserved
6	0：不记录；1：已经被记录
7	Reserved

调用示例：

2. TLIBCANFD: CANFD 总线数据类型

```
class TLIBCANFD(Structure):
    _pack_ = 1
    _fields_ = [("FIdxChn", c_uint8),
                ("FProperties", c_uint8),
                ("FDLC", c_uint8),
                ("FFDProperties", c_uint8),
                ("FIdentifier", c_int32),
                ("FTimeUs", c_ulonglong),
                ("FData", c_ubyte * 64),
                ]

    def __init__(self, FIdxChn=0, FDLC=8, FIdentifier=0x1, FProperties=1,
FFDProperties=1, FData=[]):

        self.FIdxChn = FIdxChn
        self.FDLC = FDLC
        if self.FDLC > 15:
            self.FDLC = 15
        self.FIdentifier = FIdentifier
        self.FProperties = FProperties
        self.FFDProperties = FFDProperties
        for i in range(len(FData)):
            self.FData[i] = FData[i]
```

成员：

FData：帧数据。最大长度为 64Bytes

FDLC：帧长度。

FIdentifier：帧 ID，如果为 0xFFFFFFFF，表示当前帧为错误帧

FIdxChn：帧通道，注意 CHANNEL_INDEX. CHN1 = 0,实际上是从 0 开始计算的。

FTImeUS：帧时间戳，64 位 us 级时间戳。

FFDProperties：存储 FD 相关的属性，如是否 FD 报文，发送过程中是否波特率可变。不同的字节位代表不同的属性值。

FProperties：存储 CAN 相关的属性，比如是否远程帧，是否扩展帧。

其中，两个属性字节定义如下：

【1】 FProperties：CAN 属性定义：该参数默认为 0，共八个 bits，每一个位的定义如下：

Bit	意义
0	0：Rx 接收报文；1：Tx 发送报文
1	0：data frame 数据帧；1：remote frame 远程帧
2	0：std frame 标准帧；1：extended frame 扩展帧
3-5	Reserved
6	0：不记录；1：已经被记录
7	Reserved

【2】 FDProperty: FD 属性定义：

Bit	意义
0	0：普通 CAN 报文；1：FDCAN 报文
1	0：关闭 BRS；1：开启 BRS
2	是否发生错误（ESI Flag）
3-7	Reserved

// [7-3] tbd

// [2] ESI, The E RROR S TATE I NDICATOR (ESI) flag is transmitted dominant by error active nodes, recessive by error passive nodes. ESI does not exist in CAN format frames

// [1] BRS, If the bit is transmitted recessive, the bit rate is switched from the standard bit rate of the A RBITRATION P HASE to the preconfigured alternate bit rate of the D ATA P HASE . If it is transmitted dominant, the bit rate is not switched. BRS does not exist in CAN format frames.

// [0] EDL: 0-normal CAN frame, 1-FD frame, added 2020-02-12, The E XTENDED D ATA L ENGTH (EDL) bit is recessive. It only exists in CAN FD format frames

调用示例：

3. TLIBLIN：LIN 总线数据类型

```
class TLIBLIN(Structure):
    _pack_ = 1
    _fields_ = [("FIdxChn", c_ubyte),
```

```
("FErrCode", c_ubyte),
("FProperties", c_ubyte),
("FDLC", c_uint8),
("FIdentifier", c_ubyte),
("FChecksum", c_ubyte),
("FStatus", c_ubyte),
("FTimeUs", c_ulonglong),
("FData", c_uint8 * 8),
]
```

成员：

FData：帧数据。最大程度为 8Bytes

FDLC：帧长度。

FIdentifier：帧 ID，如果为 0xFFFFFFFF，表示当前帧为错误帧

FIdxChn：帧通道，注意 CHANNEL_INDEX. CHN1 = 0,实际上是从 0 开始计算的。

FTImeUS：帧时间戳，64 位 us 级时间戳。

FProperties：存储 LIN 相关的属性，比如报文方向，是接收报文还是发送报文。

FStatus：报文状态。

FErrStatus：如果是错误帧，对应的错误类型。

其中，属性字节定义如下：

【1】 Properties：LIN 属性定义：该参数默认为 0，共八个 bits，每一个位的定义如下：

Bit	意义
0	0：Rx 接收报文；1：Tx 发送报文
1-3	Reserved
4-5	设备类型：主节点，从节点，监听节点
6	0：不记录；1：已经被记录
7	Reserved

4. 报文发送

5. 报文接收

1. **回调函数方式：**

简介：

设备接收到报文过后，把报文整理成标准的 TCAN/TCANFD 数据结构。然后调用用户注册的接收回调函数，通过参数把接收到的报文传递给调用者。libTSCAN 内部维护一个独立的线程，每当消息达到后，就会通过回调函数主动把数据传递给调用者，用户不需要主动去调用读取函数。

注册回调函数：

采用代理机制（python 里面类 C 函数指针），注册回调函数。需要注意的是，一定要先申请一个代理对象，然后把用户回调函数注册到该代理对象上，如下所示：

```
PCAN = POINTER(TLIBCAN)
if 'windows' in _os.lower():
    OnTx_RxFUNC_CAN = WINFUNCTYPE(None, PCAN)
else:
    OnTx_RxFUNC_CAN = CFUNCTYPE(None, PCAN)

PLIN = POINTER(TLIBLIN)
if 'windows' in _os.lower():
    OnTx_RxFUNC_LIN = WINFUNCTYPE(None, PLIN)
else:
    OnTx_RxFUNC_LIN = CFUNCTYPE(None, PLIN)

PCANFD = POINTER(TLIBCANFD)
if 'windows' in _os.lower():
    OnTx_RxFUNC_CANFD = WINFUNCTYPE(None, PCANFD)
else:
    OnTx_RxFUNC_CANFD = CFUNCTYPE(None, PCANFD)
```

回调函数使用

```
# 注册 can 发接
def tsapp_register_event_can(AHandle: c_size_t, ACallback:
OnTx_RxFUNC_CAN):
```

```
    """
    register can event
    Triggered when there is message transmission on the bus

    Args:
        AHandle (c_size_t): tsapp_connect retrun handle
        ACallback (OnTx_RxFUNC_CAN): function

    Returns:
        error code
    example:
        def on_can(ACAN):
            print(ACAN.contents.FData[0])

        on_can_event = OnTx_RxFUNC_CAN(on_can)
        tsapp_register_event_can(Handle,on_can_event)
    """
    r = dll.tscan_register_event_can(AHandle, ACallback)
    return r
```

2. 读取设备消息缓存的方式：

简介：

设备接收到报文过后，缓存在设备内部的 FIFO 中，外部程序调用函数接口从设备 FIFO 中把报文读取出来，FIFO 指针往后面移动；如果调用者一直不主动读取，会造成驱动内部 FIFO 溢出，最新的报文覆盖最旧的报文。

tsapp_receive_can_msgs

```
# can 报文接收
def tsapp_receive_can_msgs(AHandle: c_size_t, ACANBuffers: TLIBCAN,
ACANBufferSize: c_uint32, AChn: CHANNEL_INDEX,
                           ARxTx: READ_TX_RX_DEF):
    """
    receive can msgs

    Args:
        AHandle (c_size_t): tsapp_connect retrun handle
        ADataBuffers (TLIBCAN): can buffer
        ADataBufferSize (c_int32): can buffer size
        chn (c_int32): can channel
```

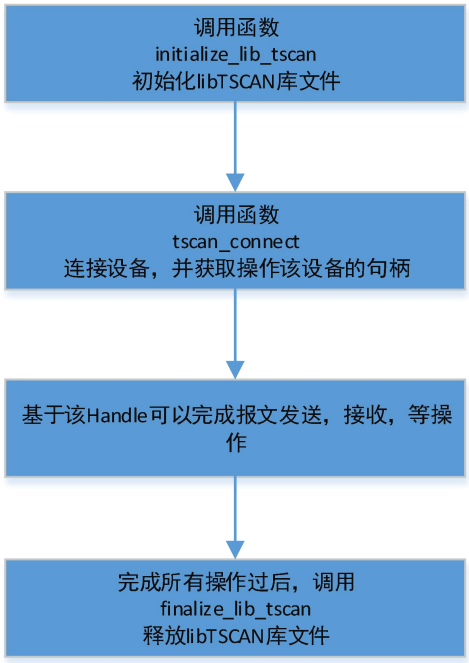
```
        ARxTx (c_int8): include tx
    Returns:
        error_code TLIBCAN_buffer TLIBCAN_bufferSize
    example:
        canbuffer = (TLIBCAN * 100)()
        size = c_int32(100)
        tsapp_receive_can_msgs(handle, canbuffer, size, 0, 1)
        for i in canbuffer:
            string = ''
            for index in range(i.FActualPayloadLength):
                string += hex(i.FData[index]) + ' '
    """
    r = dll.tsfifo_receive_can_msgs(
        AHandle, ACANBuffers, byref(ACANBufferSize), AChn, ARxTx)
    return r
```

tsfifo_canfd_receive_buffers

```
# canfd 报文接收
def tsapp_receive_canfd_msgs(AHandle: c_size_t, ACANFDBuffers:
TLIBCANFD, ACANFDBufferSize: c_uint32,
                             AChn: CHANNEL_INDEX,
                             ARxTx: READ_TX_RX_DEF):
    """
    receive canfd msgs

    Args:
        AHandle (c_size_t): tsapp_connect retrun handle
        ADataBuffers (TLIBCANFD): can buffer
        ADataBufferSize (c_int32): can buffer size
        chn (c_int32): can channel
        ARxTx (c_int8): include tx
    Returns:
        error_code TLIBCANFD_buffer TLIBCANFD_bufferSize
    example:
        canbuffer = (TLIBCANFD * 100)()
        size = c_int32(100)
        tsapp_receive_canfd_msgs(handle, canbuffer, size, 0, 1)
        for i in canbuffer:
            string = ''
            for index in range(i.FActualPayloadLength):
```

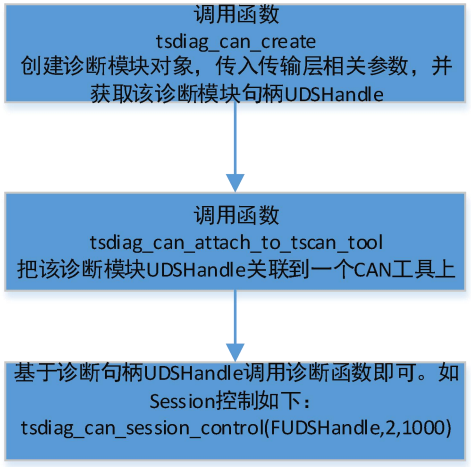
```
                string += hex(i.FData[index]) + ' '
    """
    r = dll.tsfifo_receive_canfd_msgs(
        AHandle, ACANFDBuffers, byref(ACANFDBufferSize), AChn, ARxTx)

    return r
```

6. libTSCANAPI 调用流程



7. UDS 诊断接口说明

在完成 CAN 工具其他基本配置的基础上，诊断函数使用流程如下：



8. 接口函数介绍

1. finalize_lib_tscan

函数名称	finalize_lib_tscan()
功能介绍	释放 tscan 模块
调用位置	在程序结束时自动释放它
输入参数	无
返回值	无
示例	finalize_lib_tscan()

2. initialize_lib_tsmaster

函数名称	initialize_lib_tsmaster(AEnableFIFO:c_bool,AEnableTurbe:c_bool)
功能介绍	初始化函数
调用位置	在程序加载 dll 时会自动调用
输入参数	参数 1:AEnableFIFO(c_bool)：是否使能 fifo 参数 2:AEnableTurbe(c_bool)：是否激活极速模式
返回值	无
示例	initialize_lib_tsmaster(True,True)

3. tsapp_connect

函数名称	tsapp_connect(ADeviceSerial: str, AHandle: c_size_t)
功能介绍	设备连接
调用位置	
输入参数	参数 1:ADeviceSerial (str): Equipment serial number example: b"1234568798DFE" if ADeviceSerial ==’’: Connect directly to any device 参数 2:AHandle (c_size_t): handle For specified hardware
返回值	error_code AHandle:handle For specified hardware
示例	AHandle = c_size_t(0) r = tsapp_connect(b"1234568798DFE",AHandle) or tsapp_connect("",AHandle) if(r==0 or r==5): #0 or 5 :connect success print(AHandle)）

4. tscan_scan_devices

函数名称	tscan_scan_devices(ADeviceCount: c_uint32)
功能介绍	获取设备数量
调用位置	
输入参数	参数:ADeviceCount _description_ :get devices count
返回值	error_code ADeviceCount:get devices count
示例	ADeviceCount = c_uint32(0) r = tscan_scan_devices(ADeviceCount) if r==0: #0 :get success print(ADeviceCount)

5. tscan_get_device_info

函数名称	tscan_get_device_info(ADeviceCount: c_uint32)
功能介绍	获取设备信息
调用位置	
输入参数	参数:ADeviceCount (c_uint32): hw_index
返回值	FManufacturer, FProduct, FSerial
示例	ADeviceCount = c_uint32(0) r = tscan_scan_devices(ADeviceCount) if r==0: #0 :get success for i in range(ADeviceCount): print(tscan_get_device_info(i))

6. tscan_get_error_description

函数名称	tscan_get_error_description(ACode: int)
功能介绍	获取 error_description
调用位置	
输入参数	参数: ACode (int): _description_ :error code
返回值	error code description
示例	print(tscan_get_error_description(1))

7. tsflexray_set_controller_frametrigger

函数名称	tsflexray_set_controller_frametrigger(AHandle: c_size_t, ANodeIndex:c_uint,AControllerConfig:TLibFlexray_controller_config,AFrameLengthArray:bytearray,AFrameNum:c_int,AFrameTrigger:TLibTrigger_def,AFrameTriggerNum:c_int,ATimeoutMs:c_int)
功能介绍	
调用位置	
输入参数	参数1:AHandle (c_size_t): tsapp_connect retrun handle 参数2:ANodeIndex (c_uint): flexray channle 0 or 1 参数3:AControllerConfig (TLibFlexray_controller_config): Controller Config from database config 参数4:AFrameLengthArray (bytearray): Frame Array 参数5:AFrameNum (c_int): Frame len 参数6:AFrameTrigger (TLibTrigger_def): Triggers 参数7:AFrameTriggerNum (c_int): Triggers len 参数8:ATimeoutMs (c_int): timeout
返回值	error code
示例	fr_config = TLibFlexray_controller_config(is_open_a=True, is_open_b=True, enable100_b=True, is_show_nullframe=False, is_Bridging=True) fr_trigger = (TLibTrigger_def * 3)() '''(1,0,1)''' fr_trigger[0].frame_idx = 0 fr_trigger[0].slot_id = 35 fr_trigger[0].cycle_code = 1 fr_trigger[0].config_byte = 0x33 fr_trigger[0].recv = 0 '''(3,0,4)''' fr_trigger[1].frame_idx = 1 fr_trigger[1].slot_id = 3 fr_trigger[1].cycle_code = 4 fr_trigger[1].config_byte = 0x03 fr_trigger[1].recv = 0 '''(3,3,4)''' fr_trigger[2].frame_idx = 2 fr_trigger[2].slot_id = 3 fr_trigger[2].cycle_code = 7 fr_trigger[2].config_byte = 0x03 fr_trigger[2].recv = 0 FrameLengthArray = (c_int * 3)(32, 32, 32) ret = tsflexray_set_controller_frametrigger(handle, chn0, fr_config, FrameLengthArray, 3, fr_trigger, 3, 1000)

8. tsflexray_start_net

函数名称	tsflexray_start_net(AHandle:c_size_t,ANodeIndex:c_int,ATimeoutMs: c_int)
功能介绍	开启 flexray network
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:ANodeIndex (c_int): flexray channel 参数 3:ATimeoutMs (c_int): timeout in ms
返回值	error code
示例	tsflexray_start_net(handle,0,1000)

9. tsflexray_stop_net

函数名称	tsflexray_stop_net(AHandle:c_size_t,ANodeIndex:c_int,ATimeoutMs: c_int)
功能介绍	停止 flexray network
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:ANodeIndex (c_int): flexray channel 参数 3:ATimeoutMs (c_int): timeout in ms
返回值	error code
示例	tsflexray_stop_net(handle,0,1000)

10. tsfifo_clear_flexray_receive_buffers

函数名称	tsfifo_clear_flexray_receive_buffers(AHandle: c_size_t, chn: c_int)
功能介绍	清理 flexray receive buffers
调用位置	连接硬件设备之前，先设置通道参数。
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:chn (c_int): flexray channel
返回值	error code
示例	tsfifo_clear_flexray_receive_buffers(handle,0)

11. tsflexray_transmit_async

函数名称	tsflexray_transmit_async(AHandle:c_size_t,AData:TLIBFlexray)
功能介绍	异步发送 flexray msg
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:AData (TLIBFlexray): flexray msg
返回值	error code
示例	flexray_1=TLIBFlexray(FSlotId=35,FChannelMask=1,FCycleNumber=1,FData=[1,2,3,4,5,6,7,8]) ret = tsflexray_transmit_async(handle, flexray_1)

12. tsflexray_transmit_sync

函数名称	tsflexray_transmit_sync(AHandle:c_size_t,AData:TLIBFlexray, ATimeoutMs: c_int32)
功能介绍	同步发送 flexray msg
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:AData (TLIBFlexray): flexray msg 参数 3:ATimeoutMs (c_int32):timeout
返回值	error code
示例	flexray_1=TLIBFlexray(FSlotId=35,FChannelMask=1,FCycleNumber=1,FData=[1,2,3,4,5,6,7,8]) ret = tsflexray_transmit_sync(handle,flexray_1,c_int32(100))

13. tsfifo_read_flexray_buffer_frame_count

函数名称	tsfifo_read_flexray_buffer_frame_count(AHandle:c_size_t,AIdxChn: c_int32, ACount: c_int32)
功能介绍	读取 flexray buffer frame count
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:AIdxChn (c_int32): flexray channel 参数 3:ACount (c_int32): get count
返回值	error code
示例	ACount = c_int32(0) tsfifo_read_flexray_buffer_frame_count(AHandle,0,ACount) print(ACount)

14. tsfifo_read_flexray_tx_buffer_frame_count

函数名称	tsfifo_read_flexray_tx_buffer_frame_count(AHandle:c_size_t, AIdxChn: c_int32, ACount: c_int32)
功能介绍	读取 flexray buffer tx frame count
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:AIdxChn (c_int32): flexray channel 参数 3:ACount (c_int32): get count
返回值	error code
示例	ACount = c_int32(0) tsfifo_read_flexray_tx_buffer_frame_count(AHandle,0,ACount) print(ACount)

15. tsfifo_read_flexray_rx_buffer_frame_count

函数名称	tsfifo_read_flexray_rx_buffer_frame_count(AHandle:c_size_t, AIdxChn: c_int32, ACount: c_int32)
功能介绍	读取 flexray buffer rx frame count
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:AIdxChn (c_int32): flexray channel 参数 3:ACount (c_int32): get count
返回值	error code
示例	ACount = c_int32(0) tsfifo_read_flexray_rx_buffer_frame_count(AHandle,0,ACount) print(ACount)

16. tsfifo_receive_flexray_msgs

函数名称	tsfifo_receive_flexray_msgs(AHandle: c_size_t, ADataBuffers: TLIBFlexray, ADataBufferSize: c_int32, chn: c_int32,ARxTx: c_int8)
功能介绍	接收 flexray msgs
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:ADataBuffers (TLIBFlexray): flexray buffer 参数 3:ADataBufferSize (c_int32): flexray buffer size 参数 4:chn (c_int32): flexray channel 参数 5:ARxTx (c_int8): include tx
返回值	error_code TLIBFlexray_buffer ADataBufferSize

示例	flexray_2 = (TLIBFlexray * 100)()
	size = c_int32(100)
	tsfifo_receive_flexray_msgs(handle, flexray_2, size, 0, 1)
	for i in flexray_2:
	    string = ''
	    for index in range(i.FActualPayloadLength):
	        string += hex(i.FData[index]) + ' '
	    print(i.FTimeUs, '', i.FSlotId, '', i.FCycleNumber, '', ('tx' if i.FDir else 'rx'), ″ ″, string)

17. tsapp_disconnect_by_handle

函数名称	tsapp_disconnect_by_handle(AHandle: c_size_t)
功能介绍	断开指定硬件连接
调用位置	
输入参数	参数:AHandle (c_size_t): tsapp_connect retrun handle
返回值	error code
示例	tsapp_disconnect_by_handle(handle)

18. tsapp_disconnect_all

函数名称	tsapp_disconnect_all()
功能介绍	删除所有硬件连接
调用位置	
输入参数	无
返回值	error code
示例	tsapp_disconnect_all()

19. tsapp_configure_baudrate_can

函数名称	tsapp_configure_baudrate_can(ADeviceHandle:c_size_t,AChnIdx : CHANNEL_INDEX, ARateKbps: c_double,A120: A120)
功能介绍	设置 can 参数
调用位置	
输入参数	参数 1:ADeviceHandle (c_size_t): tsapp_connect retrun handle 参数 2:AChnIdx (CHANNEL_INDEX): can channle index 参数 3:ARateKbps (c_double): baudrate 参数 4:A120 (A120): enable termination resistor
返回值	error code
示例	tsapp_configure_baudrate_can(handle,CHANNEL_INDEX.CHN1,500,A120.DEABLEA120)

20. tsapp_configure_baudrate_canfd

函数名称	tsapp_configure_baudrate_canfd(ADeviceHandle:c_size_t,AChnIdx:CHANNEL_INDEX,ARateKbps:c_double,ADataKbps:c_double,AControllerType:TLIBCANFDControllerType,AControllerMode:TLIBCANFDControllerMode,A120: A120)
功能介绍	设置 canfd 参数
调用位置	
输入参数	参数 1:ADeviceHandle (c_size_t): tsapp_connect retrun handle 参数 2:AChnIdx (CHANNEL_INDEX): chn_index 参数 3:ARateKbps (c_double): Rate baudrate 参数 4:ADataKbps (c_double): data baudrate 参数 5:AControllerType (TLIBCANFDControllerType): can isocanfd non-isocanfd 参数 6:AControllerMode (TLIBCANFDControllerMode): normol ackoff 参数 7:A120 (A120): enable termination resistor
返回值	error code
示例	tsapp_configure_baudrate_canfd(handle,CHANNEL_INDEX.CHN1,500,2000,TLIBCANFDControllerType.lfdtCAN,TLIBCANFDControllerMode.lfdmNormal,A120.A120_ENABLE)

21.tsapp_configure_can_regs

函数名称	tsapp_configure_can_regs(ADeviceHandle: c_size_t, AIdxChn: CHANNEL_INDEX, ABaudrateKbps: float, ASEG1: int,ASEG2: int, APrescaler:int,ASJ2:int,AOnlyListen:c_uint32,A120:c_uint32)
功能介绍	Can brs 采样率
调用位置	
输入参数	参数 1:ADeviceHandle (c_size_t): tsapp_connect retrun handle 参数 2:AIdxChn (CHANNEL_INDEX): chn_index 参数 3:ABaudrateKbps (float): baudrate 参数 4:ASEG1 (int): Phase buffer section1 参数 5:ASEG2 (int): Phase buffer section2 参数 6:APrescaler (int): APrescaler 参数 7:ASJ2 (int): BTL count 参数 8:AOnlyListen (c_uint32): is only listen 参数 9:A120 (c_uint32): enable termination resistor
返回值	error code
示例	tsapp_configure_can_regs(handle, CHANNEL_INDEX.CHN1, 500, 63, 16, 1, 80, 0, A120.A120_ENABLE)

22. tsapp_configure_canfd_regs

函数名称	tsapp_configure_canfd_regs(ADeviceHandle: c_size_t, AIdxChn: CHANNEL_INDEX,AArbBaudrateKbps:float,AArbSEG1:int,AArbSEG2: int,AArbPrescaler: int,AArbSJ2: int,ADataBaudrateKbps:float, ADataSEG1: int, ADataSEG2: int,ADataPrescaler: int,ADataSJ2: int,AControllerType:TLIBCANFDControllerType,AControllerMode :TLIBCANFDControllerMode,AInstallTermResistor120Ohm:c_bool)
功能介绍	canfd brs 采样率
调用位置	
输入参数	参数 1:ADeviceHandle (c_size_t): tsapp_connect retrun handle 参数 2:AIdxChn (CHANNEL_INDEX): chn_index 参数 3:AArbBaudrateKbps (float): Arbbaudrate 参数 4:AArbSEG1 (int): Arb Phase buffer section1 参数 5:AArbSEG2 (int): Arb Phase buffer section2 参数 6:AArbPrescaler (int): ArbPrescaler 参数 7:AArbSJ2 (int): Arb BTL count 参数 8:ADataBaudrateKbps (float): Databaudrate 参数 9:ADataSEG1 (int): Data Phase buffer section1 参数 10:ADataSEG2 (int): Data Phase buffer section2 参数 11:ADataPrescaler (int): Data Prescaler 参数 12:ADataSJ2 (int): Data BTL count 参数 13:AControllerType (TLIBCANFDControllerType): can isocanfd non-isocanfd 参数 14:AControllerMode (TLIBCANFDControllerMode): normol ackoff 参数 15:AInstallTermResistor120Ohm (c_bool): enable termination resistor
返回值	error code
示例	error=tsapp_configure_canfd_regs(handle,CHANNEL_INDEX.CHN1, 500,63,16,1,80,2000,63,16,1,80,TLIBCANFDControllerType.lfdtCAN,TLIBCANFDControllerMode.lfdmNormal, A120.A120_ENABLE)

23. tsapp_configure_baudrate_lin

函数名称	tsapp_configure_baudrate_lin(ADeviceHandle:c_size_t,AChnIdx : CHANNEL_INDEX, ARateKbps: c_double)
功能介绍	设置 lin 参数
调用位置	
输入参数	参数 1:ADeviceHandle (c_size_t): tsapp_connect retrun handle 参数 2:AChnIdx (CHANNEL_INDEX): lin chnidx 参数 3:ARateKbps (c_double): baudrate
返回值	error code
示例	tsapp_configure_baudrate_lin(handle,0,c_double(19.2))

24. tsapp_set_node_funtiontype

函数名称	tsapp_set_node_funtiontype(ADeviceHandle: c_size_t, AChnIdx: CHANNEL_INDEX, AFunctionType: T_LIN_NODE_FUNCTION)
功能介绍	lin 设置主节点
调用位置	
输入参数	参数 1:ADeviceHandle (c_size_t): tsapp_connect retrun handle 参数 2:AChnIdx (CHANNEL_INDEX): lin chnidx 参数 3:AFunctionType (T_LIN_NODE_FUNCTION): T_MASTER_NODE T_SLAVE_NODE
返回值	error code
示例	tsapp_set_node_funtiontype(handle,0,T_LIN_NODE_FUNCTION.T_MASTER_NODE)

25. tsapp_transmit_can_async

函数名称	tsapp_transmit_can_async(AHandle: c_size_t, Msg: TLIBCAN)
功能介绍	异步发 can 报文
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:Msg (TLIBCAN): can msg
返回值	error code
示例	msg = TLIBCAN(FIdentifier = 1,FData=[1,2,3,4,5,6,7,8]) tsapp_transmit_can_async(handle,msg)

26. tsapp_transmit_can_sync

函数名称	tsapp_transmit_can_sync(AHandle:c_size_t,Msg:TLIBCAN,ATimeoutMS: c_int32)
功能介绍	同步发 can 报文
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:Msg(TLIBCAN): can msg 参数 3:ATimeoutMS (c_int32): timeout in ms
返回值	error code
示例	msg = TLIBCAN(FIdentifier = 1,FData=[1,2,3,4,5,6,7,8]) tsapp_transmit_can_sync(handle,msg,100)

27. tsapp_transmit_canfd_async

函数名称	tsapp_transmit_canfd_async(AHandle: c_size_t, Msg:TLIBCANFD)
功能介绍	异步发 canfd 报文
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:Msg(TLIBCANFD): canfd msg
返回值	error code
示例	msg = TLIBCANFD(FIdentifier = 1,FData=[1,2,3,4,5,6,7,8]) tsapp_transmit_canfd_async(handle,msg)

28. tsapp_transmit_canfd_sync

函数名称	tsapp_transmit_canfd_sync(AHandle: c_size_t, Msg: TLIBCANFD, ATimeoutMS: c_int32)
功能介绍	同步发 canfd 报文
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:Msg(TLIBCANFD): canfd msg 参数 3:ATimeoutMS (c_int32): timeout in ms
返回值	error code
示例	msg = TLIBCANFD(FIdentifier = 1,FData=[1,2,3,4,5,6,7,8]) tsapp_transmit_canfd_sync(handle,msg,100)

29. tscan_add_cyclic_msg_can

函数名称	tscan_add_cyclic_msg_can(AHandle:c_size_t,Msg:TLIBCAN,ATimeoutMS: c_float)
功能介绍	周期发 can 报文
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:Msg(TLIBCAN): can msg 参数 3:ATimeoutMS (c_int32): timeout in ms
返回值	error code
示例	msg = TLIBCAN(FIdentifier = 1,FData=[1,2,3,4,5,6,7,8]) tscan_add_cyclic_msg_can(handle,msg,c_float(100))

30. tscan_add_cyclic_msg_canfd

函数名称	tscan_add_cyclic_msg_canfd(AHandle: c_size_t, Msg:TLIBCANFD, ATimeoutMS: c_float)
功能介绍	周期发 canfd 报文
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:Msg(TLIBCANFD): canfd msg 参数 3:ATimeoutMS (c_int32): timeout in ms
返回值	error code
示例	msg = TLIBCANFD(FIdentifier = 1,FData=[1,2,3,4,5,6,7,8]) tscan_add_cyclic_msg_canfd(handle,msg,c_float(100))

31. tscan_delete_cyclic_msg_canfd

函数名称	tscan_delete_cyclic_msg_canfd(AHandle:c_size_t,Msg:TLIBCANFD)
功能介绍	删除周期发 canfd 报文
调用位置	
输入参数	参数 1:AHandle(c_size_t): tsapp_connect retrun handle 参数 2:Msg(TLIBCANFD): canfd msg
返回值	error code
示例	msg = TLIBCANFD(FIdentifier = 1,FData=[1,2,3,4,5,6,7,8]) tscan_delete_cyclic_msg_canfd(handle,msg)

32. tscan_delete_cyclic_msg_can

函数名称	tscan_delete_cyclic_msg_can(AHandle: c_size_t, Msg: TLIBCAN)
功能介绍	删除周期发 can 报文
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:Msg(TLIBCAN): can msg
返回值	error code
示例	msg = TLIBCAN(FIdentifier = 1,FData=[1,2,3,4,5,6,7,8]) tscan_delete_cyclic_msg_can(handle,msg)

33. tsapp_transmit_lin_async

函数名称	tsapp_transmit_lin_async(AHandle: c_size_t, Msg: TLIBLIN)
功能介绍	异步发 lin 报文
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:Msg lin msg
返回值	error code
示例	msg = TLIBLIN(FIdentifier = 1,FData=[1,2,3,4,5,6,7,8]) tsapp_transmit_lin_async(handle,msg)

34. tsapp_transmit_lin_sync

函数名称	tsapp_transmit_lin_sync(AHandle:c_size_t,Msg:TLIBLIN,ATimeoutMS: c_int32)
功能介绍	同步发 lin 报文
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:Msg(TLIBLIN): lin msg 参数 3:ATimeoutMS (c_int32): timeout in ms
返回值	error code
示例	msg = TLIBLIN(FIdentifier = 1,FData=[1,2,3,4,5,6,7,8]) tsapp_transmit_lin_sync(handle,msg,100)

35. tsapp_receive_can_msgs

函数名称	tsapp_receive_can_msgs(AHandle:c_size_t,ACANBuffers:TLIBCAN,ACANBufferSize:c_uint32,AChn:CHANNEL_INDEX,ARxTx:READ_TX_RX_DEF)
功能介绍	can 报文接收
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:ACANBuffers (TLIBCAN): can buffer 参数 3:ACANBufferSize (c_int32): can buffer size 参数 4:AChn (c_int32): can channel 参数 5:ARxTx (c_int8): include tx
返回值	error_code TLIBCAN_buffer TLIBCAN_bufferSize

示例	canbuffer = (TLIBCAN * 100)()
	size = c_int32(100)
	tsapp_receive_can_msgs(handle, canbuffer, size, 0, 1)
	for i in canbuffer:
	  string = ''
	  for index in range(i.FActualPayloadLength):
	  string += hex(i.FData[index]) + ' '

36. tsapp_receive_canfd_msgs

函数名称	tsapp_receive_canfd_msgs(AHandle: c_size_t, ACANFDBuffers: TLIBCANFD, ACANFDBufferSize: c_uint32, AChn: CHANNEL_INDEX, ARxTx: READ_TX_RX_DEF)
功能介绍	canfd 报文接收
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:ACANFDBuffers (TLIBCANFD): canfd buffer 参数 3:ACANFDBufferSize (c_int32): canfd buffer size 参数 4:AChn (c_int32): canfd channel 参数 5:ARxTx (c_int8): include tx
返回值	error_code TLIBCANFD_buffer TLIBCANFD_bufferSize

示例	canbuffer = (TLIBCANFD * 100)()
	size = c_int32(100)
	tsapp_receive_canfd_msgs(handle, canbuffer, size, 0, 1)
	for i in canfdbuffer:
	  string = ''
	  for index in range(i.FActualPayloadLength):
	  string += hex(i.FData[index]) + ' '

37. tsapp_receive_lin_msgs

函数名称	tsapp_receive_lin_msgs(AHandle:c_size_t,ALINBuffers:TLIBLIN,ALINBufferSize:c_uint,AChn:CHANNEL_INDEX,ARxTx:READ_TX_RX_DEF)
功能介绍	lin 报文接收
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:ALINBuffers (TLIBLIN): lin buffer 参数 3:ALINBufferSize (c_int32): lin buffer size 参数 4:AChn (c_int32): lin channel 参数 5:ARxTx (c_int8): include tx
返回值	error_code TLIBLIN_buffer TLIBLIN_bufferSize

示例	linbuffer = (TLIBLIN * 100)()
	size = c_int32(100)
	tsapp_receive_lin_msgs(handle, linbuffer, size, 0, 1)
	for i in linbuffer:
	  string = ''
	  for index in range(i.FActualPayloadLength):
	  string += hex(i.FData[index]) + ' '

38. tsfifo_clear_can_receive_buffers

函数名称	tsfifo_clear_can_receive_buffers(AHandle:c_size_t,CHN:CHANNEL_INDEX)
功能介绍	清除 can_receive_buffers
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:CHN (CHANNEL_INDEX): can channel idnex
返回值	error code
示例	tsfifo_clear_can_receive_buffers(handle,CHANNEL_INDEX.CHN1)

39. tsfifo_clear_canfd_receive_buffers

函数名称	tsfifo_clear_canfd_receive_buffers(AHandle: c_size_t, CHN: CHANNEL_INDEX)
功能介绍	清理 canfd receive buffers
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:CHN (CHANNEL_INDEX): canfd channel idnex
返回值	error code
示例	tsfifo_clear_canfd_receive_buffers(handle,CHANNEL_INDEX.CHN1)

40. tsfifo_clear_lin_receive_buffers

函数名称	tsfifo_clear_lin_receive_buffers(AHandle:c_size_t,CHN:CHANNEL_INDEX)
功能介绍	清理 lin receive buffers
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:CHN (CHANNEL_INDEX): lin channel idnex
返回值	error code
示例	tsfifo_clear_lin_receive_buffers(handle,CHANNEL_INDEX.CHN1)

41. tslog_start

函数名称	tslog_start(AHandle: c_size_t, filePathName: str)
功能介绍	记录 can msg include canfd msg
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:filePathName (str): save log_file path_name (blf file) Absolute path
返回值	error code
示例	tslog_start(handle,Absolute path)

42. tslog_stop

函数名称	tslog_stop()
功能介绍	停止记录
调用位置	
输入参数	无
返回值	error code
示例	tslog_stop()

43.blf_to_convert

函数名称	blf_to_convert(oldpathName:str,newpathName:str,convertType:CONVERTTYPE)
功能介绍	文件转换
调用位置	
输入参数	参数 1:oldpathName (str): old log file 参数 2:newpathName (str): new log file 参数 3:convertType (CONVERTTYPE): convert type
返回值	error code
示例	blf_to_convert("1.blf","2.asc".CONVERTTYPE.ASC)

44.tslog_start_online_replay

函数名称	tslog_start_online_replay(handle:c_size_t,PathFileName:str,include_rx: bool)
功能介绍	开启在线回放
调用位置	
输入参数	参数 1:handle (c_size_t): tsapp_connect retrun handle 参数 2:PathFileName (str): blf path name Absolute path 参数 3:include_rx (bool): include rx
返回值	error code
示例	tslog_start_online_replay(handle,"/home/1.blf",False)

45.tscan_register_event_connected

函数名称	tscan_register_event_connected(ACallback: On_Connect_FUNC)
功能介绍	注册连接事件
调用位置	当设备连接成功时发生的事件
输入参数	参数:ACallback (On_Connect_FUNC): function
返回值	error code
示例	def on_connect(ps64): print("connect") on_connect_event = On_Connect_FUNC(on_connect) tscan_register_event_connected(on_connect_event)

46. tscan_register_event_disconnected

函数名称	tscan_register_event_disconnected(ACallback:On_disConnect_FUNC)
功能介绍	注册断开事件
调用位置	当设备成功断开连接时会发生
输入参数	参数 1:ACallback (On_disConnect_FUNC): function
返回值	error code
示例	def on_disconnect(ps64): print("disconnect") on_disconnect_event = On_disConnect_FUNC(on_disconnect) tscan_register_event_disconnected(on_disconnect_event)

47. tsapp_register_event_can

函数名称	tsapp_register_event_can(AHandle:c_size_t,ACallback:OnTx_RxFUNC_CAN)
功能介绍	注册 can 发接
调用位置	总线上有消息传输时触发
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:ACallback (OnTx_RxFUNC_CAN): function
返回值	error code
示例	def on_can(ACAN): print(ACAN.contents.FData[0]) on_can_event = OnTx_RxFUNC_CAN(on_can) tsapp_register_event_can(Handle,on_can_event)

48. tsapp_unregister_event_can

函数名称	tsapp_unregister_event_can(AHandle:c_size_t,ACallback:OnTx_RxFUNC_CAN)
功能介绍	注销 can 发接
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:ACallback (OnTx_RxFUNC_CAN): function
返回值	error code
示例	def on_can(ACAN): print(ACAN.contents.FData[0]) on_can_event = OnTx_RxFUNC_CAN(on_can) tsapp_unregister_event_can(Handle,on_can_event)

49. tsapp_register_pretx_event_can

函数名称	tsapp_register_pretx_event_can(AHandle: c_size_t, ACallback: OnTx_RxFUNC_CAN)
功能介绍	注册预发送事件
调用位置	发送消息将触发
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:ACallback (OnTx_RxFUNC_CAN): function
返回值	error code

示例	def on_can(ACAN):
	ACAN.contents.FData[0] = 1 #All message FData[0] will only be 1
	if ACAN.contents.FIdentifier == 1:
	ACAN.contents.FData[0] = 2 #only id=1 can message FData[0] will be 2
	
	on_can_event = OnTx_RxFUNC_CAN(on_can)
	tsapp_register_pretx_event_can(Handle,on_can_event)

50. tsapp_unregister_pretx_event_can

函数名称	tsapp_unregister_pretx_event_can(AHandle:c_size_t,ACallback: OnTx_RxFUNC_CAN)
功能介绍	注销 can 预发送事件
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:ACallback (OnTx_RxFUNC_CAN): function
返回值	error code

示例	def on_can(ACAN):
	ACAN.contents.FData[0] = 1 #All message FData[0] will only be 1
	if ACAN.contents.FIdentifier == 1:
	ACAN.contents.FData[0] = 2 #only id=1 can message FData[0] will be 2
	
	on_can_event = OnTx_RxFUNC_CAN(on_can)
	tsapp_unregister_pretx_event_can(Handle,on_can_event)

51. tsapp_register_pretx_event_canfd

函数名称	tsapp_register_pretx_event_canfd(AHandle:c_size_t,ACallback : OnTx_RxFUNC_CANFD)
功能介绍	注册 canfd 预发送事件
调用位置	发送消息将触发并可以修改消息数据
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:ACallback (OnTx_RxFUNC_CANFD): function
返回值	error code

示例	def on_can(ACAN):
	ACAN.contents.FData[0] = 1 #All message FData[0] will only be 1
	if ACAN.contents.FIdentifier == 1:
	ACAN.contents.FData[0] = 2 #only id=1 can message FData[0] will be 2
	
	on_can_event = OnTx_RxFUNC_CANFD(on_can)
	tsapp_register_pretx_event_canfd(Handle,on_can_event)

52. tsapp_unregister_pretx_event_canfd

函数名称	tsapp_unregister_pretx_event_canfd(AHandle:c_size_t,ACallback: OnTx_RxFUNC_CANFD)
功能介绍	注销 canfd 预发送事件
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:ACallback (OnTx_RxFUNC_CANFD): function
返回值	error code

示例	def on_can(ACAN):
	ACAN.contents.FData[0] = 1 #All message FData[0] will only be 1
	if ACAN.contents.FIdentifier == 1:
	ACAN.contents.FData[0] = 2 #only id=1 can message FData[0] will be 2
	
	on_can_event = OnTx_RxFUNC_CANFD(on_can)
	tsapp_unregister_pretx_event_canfd(Handle,on_can_event)

53. tsapp_register_event_canfd

函数名称	tsapp_register_event_canfd(AHandle:c_size_t,ACallback:OnTx_RxFUNC_CANFD)
功能介绍	注册 canfd 发接
调用位置	总线上有消息传输时触发
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:ACallback (OnTx_RxFUNC_CANFD): function
返回值	error code
示例	def on_can(ACAN): print(ACAN.contents.FData[0]) on_can_event = OnTx_RxFUNC_CANFD(on_can) tsapp_register_event_canfd(Handle,on_can_event)

54. tsapp_unregister_event_canfd

函数名称	tsapp_unregister_event_canfd(AHandle: c_size_t, ACallback: OnTx_RxFUNC_CANFD)
功能介绍	注销 canfd 发接
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:ACallback (OnTx_RxFUNC_CANFD): function
返回值	error code
示例	def on_can(ACAN): print(ACAN.contents.FData[0]) on_can_event = OnTx_RxFUNC_CANFD(on_can) tsapp_unregister_event_canfd(Handle,on_can_event)

55. tsapp_register_event_flexray

函数名称	tsapp_register_event_flexray(AHandle: c_size_t, ACallback: OnTx_RxFUNC_Flexray)
功能介绍	注册 flexray event
调用位置	总线上有消息传输时触发
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:ACallback (OnTx_RxFUNC_Flexray): function
返回值	error code
示例	def on_flexray(AFlexray): print(AFlexray.contents.FData[0]) on_flexray_event = OnTx_RxFUNC_Flexray(on_flexray) tsapp_register_event_flexray(Handle,on_flexray_event)

56. tsapp_unregister_event_flexray

函数名称	tsapp_unregister_event_flexray(AHandle: c_size_t, ACallback: OnTx_RxFUNC_Flexray)
功能介绍	注销 flexray 发接
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:ACallback (OnTx_RxFUNC_Flexray): function=
返回值	error code
示例	def on_flexray(AFlexray): print(AFlexray.contents.FData[0]) on_flexray_event = OnTx_RxFUNC_Flexray(on_flexray) tsapp_unregister_event_flexray(Handle,on_flexray_event)

57. tsapp_register_pretx_event_flexray

函数名称	tsapp_register_pretx_event_flexray(AHandle:c_size_t,ACallback: OnTx_RxFUNC_Flexray)
功能介绍	注册 Flexray 预发送事件
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:ACallback (OnTx_RxFUNC_Flexray): function
返回值	error code

示例	def on_flexray(AFlexray):
	    AFlexray.contents.FData[0] = 1 #All transmit tx message FData[0] will only be 1
	    if AFlexray.contents.FIdentifier == 1:
	        AFlexray.contents.FData[0] = 2 #only id=1 can message FData[0] will be 2
	
	on_flexray_event = OnTx_RxFUNC_Flexray(on_flexray)
	tsapp_register_pretx_event_flexray(Handle,on_flexray_event)

58. tsapp_unregister_pretx_event_flexray

函数名称	tsapp_unregister_pretx_event_flexray(AHandle:c_size_t,ACallback: OnTx_RxFUNC_Flexray)
功能介绍	注销 flexray 预发送事件
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:ACallback (OnTx_RxFUNC_Flexray): function
返回值	error code

示例	def on_flexray(AFlexray):
	    AFlexray.contents.FData[0] = 1 #All transmit tx message FData[0] will only be 1
	    if AFlexray.contents.FIdentifier == 1:
	        AFlexray.contents.FData[0] = 2 #only id=1 can message FData[0] will be 2
	
	on_flexray_event = OnTx_RxFUNC_Flexray(on_flexray)
	tsapp_unregister_pretx_event_flexray(Handle,on_flexray_event)

59. tsapp_register_event_lin

函数名称	tsapp_register_event_lin(AHandle:c_size_t,ACallback:OnTx_RxFUNC_LIN)
功能介绍	注册 lin 发接
调用位置	总线上有消息传输时触发
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:ACallback (OnTx_RxFUNC_LIN): function
返回值	error code
示例	def on_lin(ALIN): print(ALIN.contents.FData[0]) on_lin_event = OnTx_RxFUNC_LIN(on_lin) tsapp_register_event_lin(Handle,on_lin_event)

60. tsapp_unregister_event_lin

函数名称	tsapp_unregister_event_lin(AHandle:c_size_t,ACallback:OnTx_RxFUNC_LIN)
功能介绍	注销 lin 发接
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:ACallback (OnTx_RxFUNC_LIN): function
返回值	error code
示例	def on_lin(ALIN): print(ALIN.contents.FData[0]) on_lin_event = OnTx_RxFUNC_LIN(on_lin) tsapp_unregister_event_lin(Handle,on_lin_event)

61. tsapp_register_pretx_event_lin

函数名称	tsapp_register_pretx_event_lin(AHandle: c_size_t, ACallback: OnTx_RxFUNC_LIN)
功能介绍	注册 lin 预发送事件
调用位置	发送消息将触发并可以修改消息数据(使用 transmit_flexray 触发器)
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:ACallback (OnTx_RxFUNC_LIN): function
返回值	error code

示例	def on_lin(ALIN):
	    ALIN.contents.FData[0] = 1 #All transmit tx message FData[0] will only be 1
	    if ALIN.contents.FIdentifier == 1:
	        ALIN.contents.FData[0] = 2 #only id=1 can message FData[0] will be 2
	
	on_lin_event = OnTx_RxFUNC_LIN(on_lin)
	tsapp_register_pretx_event_lin(Handle,on_lin_event)

62. tsapp_unregister_pretx_event_lin

函数名称	tsapp_unregister_pretx_event_lin(AHandle:c_size_t,ACallback: OnTx_RxFUNC_LIN)
功能介绍	注销 lin 预发送事件
调用位置	
输入参数	参数 1:AHandle (c_size_t): tsapp_connect retrun handle 参数 2:ACallback (OnTx_RxFUNC_LIN): function
返回值	error code

示例	def on_lin(ALIN):
	    ALIN.contents.FData[0] = 1 #All transmit tx message FData[0] will only be 1
	    if ALIN.contents.FIdentifier == 1:
	        ALIN.contents.FData[0] = 2 #only id=1 can message FData[0] will be 2
	
	on_lin_event = OnTx_RxFUNC_LIN(on_lin)
	tsapp_unregister_pretx_event_lin(Handle,on_lin_event)