

TSMaster_COM接口_C#编程指导 V1.0



1. 什么情况下需要此文档.....	12
2. 使用 SDK 工程.....	12
3. 驱动特点.....	12
4. 添加 API 文件.....	12
5. 数据类型定义.....	13
6. 设备初始化.....	15
7. CAN 报文收发.....	17
8. LIN 报文收发.....	18
9. 数据库解析.....	19
10. 接口函数介绍.....	20
1. set_current_application.....	20
2. del_application.....	20
3. add_application.....	20
4. get_application_list.....	21
5. set_can_channel_count.....	21
6. set_lin_channel_count.....	21
7. get_can_channel_count.....	22
8. get_lin_channel_count.....	22
9. log.....	23
10. set_mapping.....	23
11. get_mapping.....	24
12. del_mapping.....	24

13. connect.....	24
14. disconnect.....	25
15. get_current_application.....	25
16. get_error_description.....	25
17. get_timestamp.....	26
18. configure_baudrate_can.....	26
19. configure_baudrate_canfd.....	27
20. wait.....	27
21. set_mapping_verbose.....	28
22. get_mapping_verbose.....	28
23. set_vendor_detect_preferences.....	29
24. get_vendor_detect_preferences.....	29
25. make_toast.....	30
26. get_system_var_generic.....	30
27. set_system_var_generic.....	31
28. wait_system_var.....	31
29. transmit_can_async.....	32
30. transmit_can_sync.....	32
31. transmit_canfd_async.....	33
32. transmit_canfd_sync.....	33
33. transmit_lin_async.....	34
34. transmit_lin_sync.....	35

35. add_cyclic_msg_can	35
36. add_cyclic_msg_canfd	36
37. delete_cyclic_msg_can	37
38. delete_cyclic_msg_canfd	37
39. delete_cyclic_msgs	38
40. enable_bus_statistics	38
41. clear_bus_statistics	38
42. get_bus_statistics	39
43. get_fps_can	39
44. get_fps_canfd	40
45. get_fps_lin	40
46. start_logging	41
47. stop_logging	41
48. enable_event_can	41
49. enable_event_canfd	42
50. enable_event_lin	42
51. transmit_can_async_verbose	43
52. transmit_can_sync_verbose	43
53. transmit_canfd_async_verbose	44
54. transmit_canfd_sync_verbose	44
55. transmit_lin_async_verbose	45
56. transmit_lin_sync_verbose	45

57. add_cyclic_msg_can_verbose	46
58. add_cyclic_msg_canfd_verbose	46
59. delete_cyclic_msg_can_verbose	47
60. delete_cyclic_msg_canfd_verbose	47
61. can_rbs_start	48
62. can_rbs_stop	48
63. can_rbs_is_running	48
64. can_rbs_configure	49
65. can_rbs_activate_all_networks	49
66. can_rbs_activate_network_by_name	50
67. can_rbs_activate_node_by_name	50
68. can_rbs_activate_message_by_name	51
69. can_rbs_get_signal_value_by_element	51
70. can_rbs_get_signal_value_by_address	52
71. can_rbs_set_signal_value_by_element	52
72. can_rbs_set_signal_value_by_address	53
73. fifo_enable_receive_fifo	53
74. fifo_disable_receive_fifo	53
75. fifo_enable_receive_error_frames	54
76. fifo_disable_receive_error_frames	54
77. fifo_receive_can_msg	55
78. fifo_receive_canfd_msg	55

79. fifo_receive_lin_msg	56
80. fifo_receive_fastlin_msg	57
81. fifo_clear_can_receive_buffers	57
82. fifo_clear_canfd_receive_buffers	58
83. fifo_clear_lin_receive_buffers	58
84. fifo_clear_fastlin_receive_buffers	58
85. fifo_read_can_buffer_frame_count	59
86. fifo_read_can_tx_buffer_frame_count	59
87. fifo_read_can_rx_buffer_frame_count	60
88. fifo_read_canfd_buffer_frame_count	60
89. fifo_read_canfd_tx_buffer_frame_count	61
90. fifo_read_canfd_rx_buffer_frame_count	61
91. fifo_read_lin_buffer_frame_count	61
92. fifo_read_lin_tx_buffer_frame_count	62
93. fifo_read_lin_rx_buffer_frame_count	62
94. fifo_read_fastlin_buffer_frame_count	63
95. fifo_read_fastlin_tx_buffer_frame_count	63
96. fifo_read_fastlin_rx_buffer_frame_count	63
97. flexray_rbs_start	64
98. flexray_rbs_stop	64
99. flexray_rbs_is_running	65
100. flexray_rbs_configure	65

101. flexray_rbs_activate_all_clusters	66
102. flexray_rbs_activate_cluster_by_name	66
103. flexray_rbs_activate_ecu_by_name	66
104. flexray_rbs_activate_frame_by_name	67
105. flexray_rbs_get_signal_value_by_element	67
106. flexray_rbs_get_signal_value_by_address	68
107. flexray_rbs_set_signal_value_by_element	68
108. flexray_rbs_set_signal_value_by_address	69
109. flexray_rbs_enable	69
110. flexray_rbs_batch_set_start	70
111. flexray_rbs_batch_set_end	70
112. flexray_rbs_batch_set_signal	71
113. flexray_rbs_set_frame_direction	71
114. flexray_rbs_set_normal_signal	72
115. flexray_rbs_set_rc_signal	72
116. flexray_rbs_set_rc_signal_with_limit	73
117. flexray_rbs_set_crc_signal	73
118. flexray_start_net	74
119. flexray_stop_net	74
120. transmit_lin_wakeup_async	74
121. transmit_lin_gotosleep_async	75
122. show_main_form	75

123. hide_main_form.....	76
124. load_project.....	76
125. create_project.....	76
126. save_project.....	77
127. show_tab_by_index.....	77
128. show_tab_by_name.....	78
129. get_mp_list.....	78
130. get_mp_function_list.....	78
131. get_mp_function_prototype.....	79
132. dynamic_invoke.....	79
133. load_mp.....	80
134. unload_mp.....	80
135. unload_all_mps.....	81
136. call_system_api.....	81
137. call_library_api.....	82
138. add_online_replay_config.....	82
139. set_online_replay_config.....	83
140. get_online_replay_count.....	83
141. get_online_replay_config.....	84
142. del_online_replay_config.....	85
143. del_online_replay_configs.....	85
144. start_online_replay.....	85

145. start_online_replays.....	86
146. pause_online_replay.....	86
147. pause_online_replays.....	87
148. stop_online_replay.....	87
149. stop_online_replays.....	87
150. get_online_replay_status.....	88
151. load_can_db.....	88
152. unload_can_db.....	89
153. unload_can_dbs.....	89
154. get_can_db_count.....	90
155. get_can_db_id.....	90
156. get_can_db_info.....	90
157. set_signal_value_can.....	91
158. get_signal_value_can.....	91
159. set_signal_value_canfd.....	92
160. get_signal_value_canfd.....	92
161. set_signal_value_can_verbose.....	93
162. get_signal_value_can_verbose.....	93
163. set_signal_value_canfd_verbose.....	94
164. get_signal_value_canfd_verbose.....	95
165. load_flexray_db.....	95
166. unload_flexray_db.....	96

167. unload_flexray_dbs.....	96
168. get_flexray_db_count.....	96
169. get_flexray_db_properties_by_address.....	97
170. get_flexray_db_properties_by_address.....	98
171. get_flexray_ecu_properties_by_address.....	98
172. get_flexray_ecu_properties_by_index.....	99
173. get_flexray_frame_properties_by_address.....	99
174. get_flexray_frame_properties_by_index.....	100
175. get_flexray_signal_properties_by_address.....	101
176. get_flexray_signal_properties_by_index.....	103
177. get_flexray_db_id.....	104
178. diag_can_create.....	104
179. diag_set_fdmode.....	105
180. diag_can_delete.....	105
181. diag_can_delete_all.....	106
182. tp_can_request_and_get_response.....	106
183. tp_can_request_and_get_response_functional.....	106

1. 什么情况下需要此文档

用户基于 DotNet 平台的编程语言, 对目前市面上主流的如 Vector, TOSUN, Peak, 英特佩斯等工具进行二次开发的时候, 需要参考本文档, 调用 com 接口来实现对设备的程序控制。

2. 使用 SDK 工程

3. 驱动特点

4. 添加 API 文件

5. 数据类型定义

1.TCAN 类型:CAN 报文结构体

成员:

FIdxChn:帧通道,从 0 开始计算。

FIsTX:报文方向,0 是接收,1 是发送。

FIsRemote:远程帧,0 是数据帧,1 是远程帧。

FIsExtendedId:扩展帧,0 是标准帧,1 是扩展帧。

FIsError:错误帧,1 是错误帧。

FDLC:帧长度。

FIdentifier:帧 ID。

FTimeUs:帧时间戳,64us 级时间戳。

FData: 帧数据,最大长度为 8Bytes

2.TCANFD 类型:CANFD 报文结构体

成员:

FIdxChn:帧通道,从 0 开始计算。

FIsTX:报文方向,0 是接收,1 是发送。

FIsExtendedId:扩展帧,0 是标准帧,1 是扩展帧。

FIsError:错误帧,1 是错误帧。

FIsEDL:CANFD 帧,0 是标准 CAN 帧,1 是 CANFD 帧。

FlsBRS:BRS 位。

FlsESI:ESI 位。

FDLC:帧长度。

FIdentifier:帧 ID。

FTimeUs:帧时间戳,64us 级时间戳。

FData: 帧数据,最大长度为 64Bytes

3.TLIN 类型:LIN 报文结构体

成员:

FIdxChn:帧通道,从 0 开始计算。

FlsTX:报文方向,0 是接收,1 是发送。

FlsError:错误帧,1 是错误帧。

FDLC:帧长度。

FIdentifier:帧 ID。

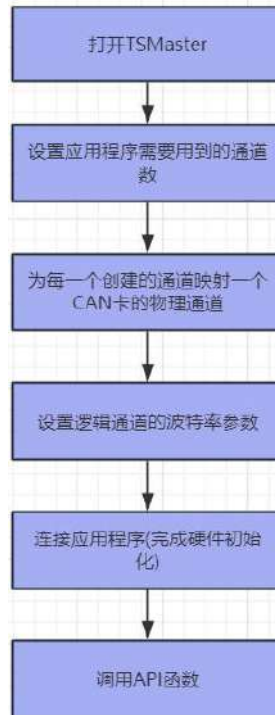
FChecksum:效验和。

FTimeUs:帧时间戳,64us 级时间戳。

FData: 帧数据,最大长度为 8Bytes

6. 设备初始化

1.CAN 初始化流



CAN 初始化示例代码:

```
TSApplicationClass app = new TSApplicationClass();//打开 TSMaster
TSCOM com = app.TSCOM();
app.set_can_channel_count(2);//设置 CAN 通道数
TTSMapping map = new TTSMapping();
map.FAppName = "COMTSMaster";//应用程序名
map.FAppChannelIndex = channel1;//映射通道
map.FAppChannelType = TTSAppChannelType.APP_CAN; //通道类型
map.FHWDeviceType = TTSBusToolDeviceType.TS_USB_DEVICE;//硬件类型
map.FHWDeviceSubType = 11; //具体型号
map.FHWIndex = 0; // 硬件序号
map.FHWChannelIndex = 0;//硬件设备通道
app.set_mapping(map);//映射通道
map.FAppChannelIndex = channel2;
map.FHWChannelIndex = 1;
app.set_mapping(map);
app.configure_baudrate_canfd(channel1, 500, 2000, TTSCANFDControllerType.lfdtISOCAN,
    TTSCANFDControllerMode.lfdmNormal, true);
app.configure_baudrate_canfd(channel2, 500, 2000, TTSCANFDControllerType.lfdtISOCAN,
```

```
TTSCANFDControllerMode.lfdmNormal, true);  
app.connect();//连接设备
```

2.LIN 初始化流



LIN 初始化示例代码:

```
TSAApplicationClass app = new TSAApplicationClass();//打开 TSMaster  
TSCOM com = app.TSCOM();  
app.set_lin_channel_count(2);//设置 LIN 通道  
TTSMapping map = new TTSMapping();  
map.FAppName = "COMTSMaster";//应用程序名  
map.FAppChannelIndex = channel1;//映射通道  
map.FAppChannelType = TTSAppChannelType.APP_LIN; //通道类型  
map.FHWDeviceType = TTSBusToolDeviceType.TS_USB_DEVICE;//硬件类型  
map.FHWDeviceSubType = 11; //具体型号  
map.FHWIndex = 0; // 硬件序号  
map.FHWChannelIndex = 0;//硬件设备通道  
app.set_mapping(map);//映射通道  
map.FAppChannelIndex = channel2;  
map.FHWChannelIndex = 1;  
app.set_mapping(map);  
app.connect();//连接设备
```


7. CAN 报文收发

1. 报文发送

CAN 报文发送代码:

```
TSApplicationClass app = new TSApplicationClass();//打开 TSMaster
TSCOM com = app.TSCOM();
TCAN tcan = new TCAN();
tcan.FIsExtendedId = 0;
tcan.FIdxChn = 0;
tcan.FIdentifier = 0x123;
tcan.FDLC = 8;
tcan.FIsRemote = 0;
tcan.FIsTX = 1;

sbyte[] data = new sbyte[8] {1, 2, 3, 4, 5, 6, 7, 8};
tcan.FDdatas = data;
com.transmit_can_async(ref tcan);
```

CANFD 报文发送代码:

```
TSApplicationClass app = new TSApplicationClass();//打开 TSMaster
TSCOM com = app.TSCOM();
TCANFD tcanfd = new TCANFD();
tcanfd.FIdxChn = 0;
tcanfd.FIdentifier = 0x6;
tcanfd.FDLC = 15;
tcanfd.FIsTX = 1;
tcanfd.FIsEDL = 1;

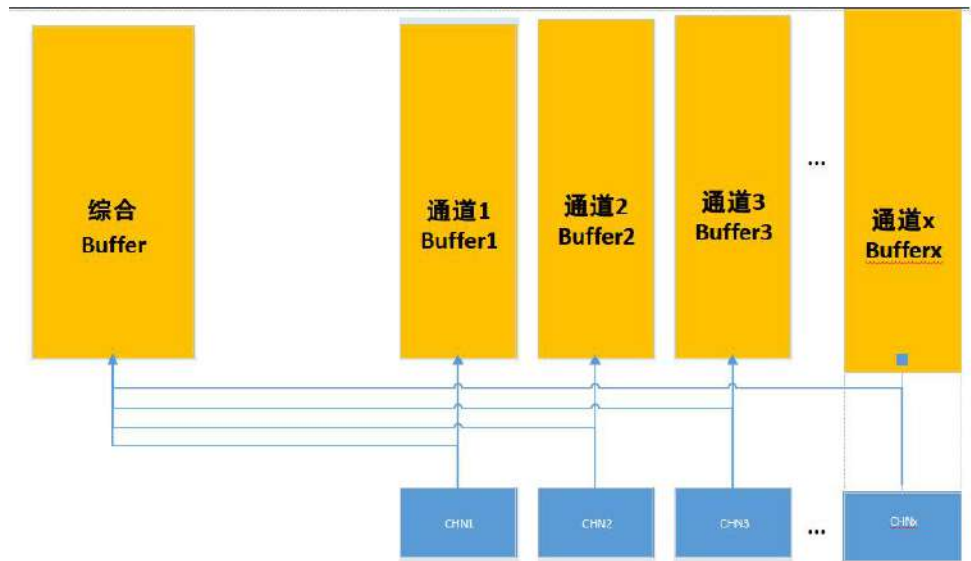
sbyte[] data = new sbyte[64];
tcanfd.FDdatas = data;
com.transmit_canfd_async(ref tcanfd);
```

2. 报文接收

读取设备消息缓存的方式:

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。TSMaster API 内部，报文缓存机制如

下图所示:



综合 FIFO: 所有通道的报文根据接收顺序放在里面。

特点: 可以看到不同通道报文的相对接收顺序。报文在里面按照接收顺序存放, 最新的报文覆盖最旧接收的报文。

通道 FIFO: 每一个通道有一个自己单独的报文 FIFO。

特点: 专用于存储跟本通道相关的报文, 通道之间互不干扰。报文在里面按照接收顺序存放, 最新的报文覆盖最旧接收的报文。

TSMaster 提供了报文读取驱动。可以选择读取指定通道的报文集合, 也可以读取综合报文里面的报文集合。

8. LIN 报文收发

1. 报文发送

LIN 报文发送代码:

```
TSApplicationClass app = new TSApplicationClass();//打开 TSMaster
TSCOM com = app.TSCOM();
TLIN tlin = new TLIN();
tlin.FIdxChn = 0;
tlin.FIdentifier = 0x12;
tlin.FIsTX = 1;
tlin.FDLC = 8;
sbyte[] data = new sbyte[8] {1, 2, 3, 4, 5, 6, 7, 8};
tlin.FDdatas = data;

com.transmit_lin_async(ref tlin);
```

9. 数据库解析

10. 接口函数介绍

1.set_current_application

函数名称	void set_current_application(string AAppName)
功能介绍	设置应用程序名
调用位置	软件连接之前
输入参数	AAppName:应用程序名
返回值	无
示例	<pre>TSApplicationClass app = new TSApplicationClass(); app.set_current_application("COMTSMaster");</pre>

2.del_application

函数名称	void del_application(string AAppName)
功能介绍	删除应用程序
调用位置	软件连接之前
输入参数	AAppName:应用程序名
返回值	无
示例	<pre>TSApplicationClass app = new TSApplicationClass(); app.del_application("COMTSMaster");</pre>

3.add_application

函数名称	void add_application(string AAppName)
功能介绍	添加应用程序
调用位置	软件连接之前

输入参数	AAppName:应用程序名
返回值	无
示例	<pre>TSApplicationClass app = new TSApplicationClass(); app.add_application("COMTSMaster");</pre>

4.get_application_list

函数名称	string get_application_list();
功能介绍	获取应用程序列表
调用位置	
输入参数	无
返回值	应用程序列表字符串
示例	<pre>string appList; TSApplicationClass app = new TSApplicationClass(); appList = app.get_application_list();</pre>

5.set_can_channel_count

函数名称	void set_can_channel_count(int ACount);
功能介绍	设置 CAN 通道数
调用位置	TSMaster 连接前
输入参数	ACount:CAN 通道数
返回值	无
示例	<pre>TSApplicationClass app = new TSApplicationClass(); app.set_can_channel_count(2);</pre>

6.set_lin_channel_count

函数名称	void set_lin_channel_count(int ACount);
------	---

功能介绍	设置 LIN 通道数
调用位置	TSMaster 连接前
输入参数	ACount:LIN 通道数
返回值	无
示例	<pre>TSApplicationClass app = new TSApplicationClass(); app.set_lin_channel_count(2);</pre>

7.get_can_channel_count

函数名称	<code>int get_can_channel_count();</code>
功能介绍	获取 CAN 通道数
调用位置	
输入参数	无
返回值	CAN 通道数
示例	<pre>TSApplicationClass app = new TSApplicationClass(); int count = app.get_can_channel_count();</pre>

8.get_lin_channel_count

函数名称	<code>int get_lin_channel_count();</code>
功能介绍	获取 LIN 通道数
调用位置	
输入参数	无
返回值	LIN 通道数
示例	<pre>TSApplicationClass app = new TSApplicationClass(); int count = app.get_lin_channel_count();</pre>

9.log

函数名称	<code>void log(string AMsg, int ALevel);</code>
功能介绍	向 TSMaster 打印系统消息
调用位置	
输入参数	AMsg:消息字符串 ALevel:消息等级 可以在 TSMaster 系统常量里面查看
返回值	无
示例	<code>TSAApplicationClass app = new TSAApplicationClass(); app.log("程序启动",3);</code>

10.set_mapping

函数名称	<code>void set_mapping(TTSMMapping* AMapping);</code>
功能介绍	通道映射
调用位置	TSMaster 连接之前
输入参数	AMapping:通道映射结构体
返回值	无
示例	<code>TTSMMapping a = new TTSMMapping(); a.FAppName = "COMTSMaster";//应用程序名 a.FAppChannelIndex = 0;//映射通道 a.FAppChannelType = TTSAAppChannelType.APP_CAN; //通道类型 a.FHWDeviceType = TTSToolDeviceType.TS_USB_DEVICE;//硬件类型 a.FHWDeviceSubType = 11; //具体型号 a.FHWIndex = 0; // 硬件序号 a.FHWChannelIndex = 0;//硬件设备通道 TSAApplicationClass app = new TSAApplicationClass(); app.set_mapping(a);//映射通道</code>

11.get_mapping

函数名称	<code>TTSMapping get_mapping(TTSAAppChannelType AAppChannelType, int AAppChannelIndex);</code>
功能介绍	获取通道映射
调用位置	
输入参数	AAppChannelType:通道类型 AAppChannelIndex:通道索引
返回值	通道映射结构体
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TTSMapping a = new TTSMapping(); a = app.get_mapping(TTSAAppChannelType.APP_CAN, 0); </pre>

12.del_mapping

函数名称	<code>void del_mapping(TTSAAppChannelType AAppChannelType,int AAppChannelIndex);</code>
功能介绍	删除映射
调用位置	
输入参数	AAppChannelType:通道类型 AAppChannelIndex:通道索引
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); app.del_mapping(TTSAAppChannelType.APP_CAN, 0); </pre>

13.connect

函数名称	<code>void connect();</code>
功能介绍	完成各个硬件参数的设置过后，调用此函数完成设备的连接。后面就可以调用设备完成数据收发等功能了。
调用位置	
输入参数	无

返回值	无
示例	<code>TSApplicationClass app = new TSApplicationClass(); app.connect();</code>

14.disconnect

函数名称	<code>void disconnect();</code>
功能介绍	断开设备连接
调用位置	不需要使用硬件设备，调用此函数断开设备连接
输入参数	无
返回值	无
示例	<code>TSApplicationClass app = new TSApplicationClass(); app.disconnect();</code>

15.get_current_application

函数名称	<code>string get_current_application();</code>
功能介绍	获取当前应用程序名
调用位置	
输入参数	无
返回值	当前应用程序名
示例	<code>TSApplicationClass app = new TSApplicationClass(); string name = app.get_current_application();</code>

16.get_error_description

函数名称	<code>string get_error_description(int AErrorCode);</code>
功能介绍	获取错误码含义

调用位置	
输入参数	AErrorCode:错误码
返回值	错误码含义
示例	<pre>TSApplicationClass app = new TSApplicationClass(); string error = app.get_error_description(97);</pre>

17.get_timestamp

函数名称	long get_timestamp();
功能介绍	获取当前测量的 PC 时间戳 us
调用位置	
输入参数	无
返回值	时间戳 us
示例	<pre>TSApplicationClass app = new TSApplicationClass(); long timestamp = app.get_timestamp();</pre>

18.configure_baudrate_can

函数名称	void configure_baudrate_can(int AldxChn,float ABaudrateKbps,bool AListenOnly, bool AInstallTermResistor120Ohm);
功能介绍	配置 CAN 通道的波特率等硬件参数
调用位置	连接 CAN 工具之前, 先调用此函数配置硬件设备参数
输入参数	AldxChn:通道索引 ABaudrateKbps:波特率 AListenOnly:是否只听模式 AInstallTermResistor120Ohm:是否使能内部终端电阻
返回值	无
示例	<pre>TSApplicationClass app = new TSApplicationClass(); app.configure_baudrate_can(0,500,false,true);</pre>

19.configure_baudrate_canfd

函数名称	<code>void configure_baudrate_canfd(int AldxChn,int AArbRateKbps, int ADataRateKbps,TTSCANFDControllerType AControllerType, TTSCANFDControllerMode AControllerMode, bool AInstallTermResistor120Ohm);</code>
功能介绍	配置 CANFD 通道的波特率等硬件参数
调用位置	连接 CAN 工具之前, 先调用此函数配置硬件设备参数
输入参数	AldxChn:通道索引 AArbRateKbps:仲裁段波特率 ADataRateKbps:数据段波特率 AControllerType:CAN 控制器类型 AControllerMode:控制器模式 AInstallTermResistor120Ohm:是否使能内部终端电阻
返回值	无
示例	<pre>TSApplicationClass app = new TSApplicationClass(); app.configure_baudrate_canfd(0,500,2000, TTSCANFDControllerType.IfdtISOCAN, TTSCANFDControllerMode.IfdmNormal, true);</pre>

20.wait

函数名称	<code>void wait(int ATimeMs);</code>
功能介绍	等待特定时间
调用位置	
输入参数	ATimeMs:等待时间 ms
返回值	无
示例	<pre>TSApplicationClass app = new TSApplicationClass(); app.log("开始等待", 3); app.wait(1000); app.log("结束等待", 3);</pre>

21.set_mapping_verbose

函数名称	void set_mapping_verbose(int AAppChannelIndex, TTSAppChannelType AAppChannelType, TTSToolDeviceType AHWDeviceType,int AHWIndex, int AHWChannelIndex,int AHWDeviceSubType, string AHWDeviceName,bool AMappingDisabled);
功能介绍	通道映射
调用位置	TSMaster 连接之前
输入参数	AAppChannelIndex:通道索引 AAppChannelType:通道类型 AHWDeviceType:硬件类型 AHWIndex:硬件序号 区分多个同类型设备 AHWChannelIndex:硬件设备通道 AHWDeviceSubType:具体型号枚举值 AHWDeviceName:设备名 AMappingDisabled:是否禁止映射
返回值	无
示例	<pre>TSApplicationClass app = new TSApplicationClass(); app.set_mapping_verbose(0,TTSAppChannelType.APP_CAN, TTSToolDeviceType.TS_USB_DEVICE,0,0,11,"TC1016",false);</pre>

22.get_mapping_verbose

函数名称	void get_mapping_verbose(TTSAppChannelType AAppChannelType, int AAppChannelIndex, out TTSToolDeviceType AHWDeviceType, out int AHWIndex, out int AHWChannelIndex, out int AHWDeviceSubType,out string AHWDeviceName, out bool AMappingDisabled);
功能介绍	获取通道映射
调用位置	
输入参数	AAppChannelType:通道类型 AAppChannelIndex:通道索引 AHWDeviceType:硬件类型 AHWIndex:硬件序号 区分多个同类型设备 AHWChannelIndex:硬件设备通道 AHWDeviceSubType:具体型号枚举值

	AHWDeviceName:设备名 AMappingDisabled:是否禁止映射
返回值	无
示例	<pre> TTSBusToolDeviceType deviceType; int hwIndex; int hwChannelIndex; int hwDeviceSubType; string hwDeviceName; bool mappingDisable; TSApplicationClass app = new TSApplicationClass(); app.get_mapping_verbose(TTSAppChannelType.APP_CAN,0,out deviceType, out hwIndex, out hwChannelIndex, out hwDeviceSubType, out hwDeviceName, out mappingDisable); </pre>

23.set_vendor_detect_preferences

函数名称	void set_vendor_detect_preferences(bool ADetectTOSUN, bool ADetectVector,bool ADetectPEAK,bool ADetectKvaser,bool ADetectZLG,bool ADetectIntrepidcs,bool ADetectCANable);
功能介绍	设置供应商检测选项,连接其他厂家的硬件,需要先调用此函数
调用位置	通道映射之前
输入参数	ADetectTOSUN: 同星设备 ADetectVector:Vector 设备 ADetectPEAK:PEAK 设备 ADetectKvaser:Kvaser 设备 ADetectZLG:ZLG 设备 ADetectIntrepidcs:Intrepidcs 设备 ADetectCANable:CANable 设备
返回值	无
示例	<pre> TSApplicationClass app = new TSApplicationClass(); app.set_vendor_detect_preferences(true, true, false, false, false, false, false); </pre>

24.get_vendor_detect_preferences

函数名称	void get_vendor_detect_preferences(out bool ADetectTOSUN, out bool ADetectVector, out bool ADetectPEAK, out bool ADetectKvaser, out bool ADetectZLG,
------	--

	<code>out bool ADetectIntrepidcs, out bool ADetectCANable);</code>
功能介绍	获取供应商检测选项
调用位置	
输入参数	ADetectTOSUN: 同星设备 ADetectVector:Vector 设备 ADetectPEAK:PEAK 设备 ADetectKvaser:Kvaser 设备 ADetectZLG:ZLG 设备 ADetectIntrepidcs:Intrepidcs 设备 ADetectCANable:CANable 设备
返回值	无
示例	<pre>bool tosun, vector, peak, kvaser,zlg,intrepidcs,canable; TSApplicationClass app = new TSApplicationClass(); app.get_vendor_detect_preferences(out tosun, out vector, out peak, out kvaser, out zlg, out intrepidcs, out canable);</pre>

25.make_toast

函数名称	<code>void make_toast(string AString,int ALevel);</code>
功能介绍	TSMaster 弹出提示窗,同时系统消息打印信息
调用位置	
输入参数	AMsg:打印消息 ALevel:消息等级 可以在 TSMaster 系统常量里面查看
返回值	无
示例	<pre>TSApplicationClass app = new TSApplicationClass(); app.make_toast("测试消息", 3);</pre>

26.get_system_var_generic

函数名称	<code>bool get_system_var_generic(string ACompleteName, out string AValue);</code>
功能介绍	获取系统变量值
调用位置	

输入参数	ACompleteName:系统变量名 AValue:系统变量值
返回值	函数执行结果 true:成功
示例	<pre>bool tosun, vector, peak, kvaser,zlg,intrepidcs,canable; TSApplicationClass app = new TSApplicationClass(); app.get_vendor_detect_preferences(out tosun, out vector, out peak, out kvaser, out zlg, out intrepidcs, out canable);</pre>

27.set_system_var_generic

函数名称	bool set_system_var_generic(string ACompleteName, string AValue);
功能介绍	设置系统变量值
调用位置	
输入参数	ACompleteName:系统变量名 AValue:系统变量值
返回值	函数执行结果 true:成功
示例	<pre>TSApplicationClass app = new TSApplicationClass(); if(app.set_system_var_generic("Var0", "4")) app.log("设置系统变量成功" ,3);</pre>

28.wait_system_var

函数名称	bool wait_system_var(string ACompleteName, string AValue, int ATimeoutMs);
功能介绍	判断一定时间内,系统变量是否等于指定值
调用位置	
输入参数	ACompleteName:系统变量名 AValue:系统变量值 ATimeoutMs:超时时间
返回值	判断结果
示例	<pre>TSApplicationClass app = new TSApplicationClass(); if (app.wait_system_var("Var0", "5", 5000)) app.log("等于指定值",3); else</pre>

	app.log("不等于指定值",3);
--	----------------------

29.transmit_can_async

函数名称	void transmit_can_async(ref TCAN ACAN);
功能介绍	异步发送 CAN 报文
调用位置	设备连接后
输入参数	ACAN:CAN 报文结构体
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); app.connect(); TCAN tcan = new TCAN(); tcan.FlsExtendedId = 0; tcan.FIdxChn = 0; tcan.FIdentifier = 0x123; tcan.FDLC = 8; tcan.FlsRemote = 0; sbyte[] data = new sbyte[8] {1,2,3,4,5,6,7,8}; tcan.FDdatas = data; com.transmit_can_async(ref tcan); </pre>

30.transmit_can_sync

函数名称	void transmit_can_sync(ref TCAN ACAN, int ATimeoutMs, out bool ASuccess);
功能介绍	同步发送 CAN 报文
调用位置	设备连接后
输入参数	ACAN:CAN 报文结构体 ATimeoutMs:超时时间 ms ASuccess:报文发送是否成功
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); </pre>

	<pre> app.connect(); TCAN tcan = new TCAN(); tcan.FlsExtendedId = 0; tcan.FIdxChn = 0; tcan.FIdentifier = 0x123; tcan.FDLC = 8; tcan.FlsRemote = 0; sbyte[] data = new sbyte[8] {1,2,3,4,5,6,7,8}; tcan.FDdatas = data; bool success = false; com.transmit_can_sync(ref tcan,100,out success); if (success) app.log("报文发送成功",3); </pre>
--	---

31.transmit_canfd_async

函数名称	void transmit_canfd_async(ref TCANFD ACANFD);
功能介绍	异步发送 CANFD 报文
调用位置	设备连接后
输入参数	ACANFD:CANFD 报文结构体
返回值	无
示例	<pre> TSApplicationClass app = new TSApplicationClass(); TSCOM com = app.TSCOM(); app.connect(); TCANFD tcan = new TCANFD(); tcan.FIdxChn = 0; tcan.FIdentifier = 0x123; tcan.FDLC = 15; tcan.FlsExtendedId = 0; tcan.FlsEDL = 1; sbyte[] data = new sbyte[64]; tcan.FDdatas = data; com.transmit_canfd_async(ref tcan); </pre>

32.transmit_canfd_sync

函数名称	void transmit_canfd_sync(ref TCANFD ACANFD,int ATimeoutMs, out bool ASuccess);
------	--

功能介绍	同步发送 CANFD 报文
调用位置	设备连接后
输入参数	ACANFD:CANFD 报文结构体 ATimeoutMs:超时时间 ASuccess:报文发送是否成功
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); app.connect(); TCANFD tcan = new TCANFD(); tcan.FIdxChn = 0; tcan.FIdentifier = 0x123; tcan.FDLC = 15; tcan.FIsExtendedId = 0; tcan.FIsEDL = 1; sbyte[] data = new sbyte[64]; tcan.FDatas = data; bool success = false; com.transmit_canfd_sync(ref tcan,100,out success); if (success) app.log("报文发送成功", 3); </pre>

33.transmit_lin_async

函数名称	void transmit_lin_async(ref TLIN ALIN);
功能介绍	异步发送 LIN 报文
调用位置	设备连接后
输入参数	ALIN:LIN 报文结构体
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); app.connect(); TLIN tlin = new TLIN(); tlin.FIdxChn = 0; tlin.FIdentifier = 0X1; </pre>

	<pre> tlin.FDLC = 0x8; tlin.FIsTX = 1; sbyte[] data = new sbyte[8]; tlin.FDdatas = data; com.transmit_lin_async(ref tlin); </pre>
--	---

34.transmit_lin_sync

函数名称	void transmit_lin_sync(ref TLIN ALIN, int ATimeoutMs, out bool ASuccess);
功能介绍	同步发送 LIN 报文
调用位置	设备连接后
输入参数	ALIN:LIN 报文结构体 ATimeoutMs:超时时间 ASuccess:报文发送是否成功
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); app.connect(); TLIN tlin = new TLIN(); tlin.FIdxChn = 0; tlin.FIdentifier = 0X1; tlin.FDLC = 0x8; tlin.FIsTX = 1; sbyte[] data = new sbyte[8]; tlin.FDdatas = data; bool success = false; com.transmit_lin_sync(ref tlin,100,out success); if (success) app.log("报文发送成功", 3); </pre>

35.add_cyclic_msg_can

函数名称	void add_cyclic_msg_can(ref TCAN ACAN, float APeriodMs);
功能介绍	周期发送 CAN 报文
调用位置	设备连接后

输入参数	ACAN:CAN 报文结构体 APeriodMs:周期时间 ms
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); app.connect(); TCAN tcan = new TCAN(); tcan.FIdxChn = 0; tcan.FIdentifier = 0X1; tcan.FDLC = 0x8; sbyte[] data = new sbyte[8]; tcan.FDdatas = data; com.add_cyclic_msg_can(ref tcan,1000); </pre>

36.add_cyclic_msg_canfd

函数名称	void add_cyclic_msg_canfd(ref TCANFD ACANFD,float APeriodMs);
功能介绍	周期发送 CANFD 报文
调用位置	设备连接后
输入参数	ACANFD:CANFD 报文结构体 APeriodMs:周期时间 ms
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); app.connect(); TCANFD tcan = new TCANFD(); tcan.FIdxChn = 0; tcan.FIdentifier = 0X1; tcan.FIsEDL = 1; tcan.FDLC = 15; sbyte[] data = new sbyte[64]; tcan.FDdatas = data; com.add_cyclic_msg_canfd(ref tcan,1000); </pre>

37.delete_cyclic_msg_can

函数名称	void delete_cyclic_msg_can(ref TCAN ACAN);
功能介绍	停止周期发送 CAN 报文
调用位置	设备连接后
输入参数	ACAN:CAN 报文结构体
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); app.connect(); TCAN tcan = new TCAN(); tcan.FIdxChn = 0; tcan.FIdentifier = 0X1; tcan.FDLC = 8; sbyte[] data = new sbyte[8]; tcan.FDdatas = data; com.delete_cyclic_msg_can(ref tcan); </pre>

38.delete_cyclic_msg_canfd

函数名称	void delete_cyclic_msg_canfd(ref TCANFD ACANFD);
功能介绍	停止周期发送 CANFD 报文
调用位置	设备连接后
输入参数	ACANFD:CANFD 报文结构体
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); app.connect(); TCANFD tcan = new TCANFD(); tcan.FIdxChn = 0; tcan.FIdentifier = 0X1; tcan.FIsEDL = 1; </pre>

	<pre>tcan.FDLC = 15; sbyte[] data = new sbyte[64]; tcan.FDdatas = data; com.delete_cyclic_msg_canfd(ref tcan);</pre>
--	--

39.delete_cyclic_msgs

函数名称	void delete_cyclic_msgs();
功能介绍	停止所有周期发送报文
调用位置	设备连接后
输入参数	无
返回值	无
示例	<pre>TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.delete_cyclic_msgs();</pre>

40.enable_bus_statistics

函数名称	void enable_bus_statistics(bool AEnable);
功能介绍	启动总线统计信息
调用位置	
输入参数	AEnable:是否启动
返回值	无
示例	<pre>TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.enable_bus_statistics(true);</pre>

41.clear_bus_statistics

函数名称	void clear_bus_statistics();
------	------------------------------

功能介绍	清除总线统计信息
调用位置	
输入参数	无
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.clear_bus_statistics(); </pre>

42.get_bus_statistics

函数名称	<pre> double get_bus_statistics(TTAppChannelType ABusType,int AldxChn, TTSCANBusStatistics AldxStat); </pre>
功能介绍	获取总线统计信息
调用位置	
输入参数	ABusType:通道类型 AldxChn:通道索引 AldxStat:总线统计类型
返回值	统计值
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.get_bus_statistics(TTAppChannelType.APP_CAN, 0, TTSCANBusStatistics.cbsBusLoad); </pre>

43.get_fps_can

函数名称	<pre> int get_fps_can(int AldxChn,int AIdentifier); </pre>
功能介绍	获取 CAN 报文帧率
调用位置	
输入参数	AldxChn:CAN 通道 AIdentifier:报文 id
返回值	帧率

示例	<pre>int fps; TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); fps = com.get_fps_can(0, 0x123);</pre>
----	--

44.get_fps_canfd

函数名称	int get_fps_canfd(int AldxChn,int AIdentifier);
功能介绍	获取 CANFD 报文帧率
调用位置	
输入参数	AldxChn:CAN 通道 AIdentifier:报文 id
返回值	帧率
示例	<pre>int fps; TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); fps = com.get_fps_canfd(0, 0x123);</pre>

45.get_fps_lin

函数名称	int get_fps_lin(int AldxChn,int AIdentifier);
功能介绍	获取 LIN 报文帧率
调用位置	
输入参数	AldxChn:LIN 通道 AIdentifier:报文 id
返回值	帧率
示例	<pre>int fps; TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); fps = com.get_fps_lin(0, 0x1);</pre>

46.start_logging

函数名称	void start_logging(string AFileName);
功能介绍	开始总线记录
调用位置	
输入参数	AFileName:blf 文件保存路径
返回值	无
示例	<pre>TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.start_logging("C:/Users/LIN/Desktop/test/abc.blf");</pre>

47.stop_logging

函数名称	void stop_logging();
功能介绍	结束总线记录
调用位置	
输入参数	无
返回值	无
示例	<pre>TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.stop_logging();</pre>

48.enable_event_can

函数名称	void enable_event_can(bool AEnable);
功能介绍	使能 CAN 事件
调用位置	

输入参数	AEnable:使能
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.enable_event_can(true); </pre>

49.enable_event_canfd

函数名称	void enable_event_canfd(bool AEnable);
功能介绍	使能 CANFD 事件
调用位置	
输入参数	AEnable:使能
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.enable_event_canfd(true); </pre>

50.enable_event_lin

函数名称	void enable_event_lin(bool AEnable);
功能介绍	使能 LIN 事件
调用位置	
输入参数	AEnable:使能
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.enable_event_lin(true); </pre>

51.transmit_can_async_verbose

函数名称	<code>void transmit_can_async_verbose(int AldxChn,bool AlsRemote, bool AlsExtended,int ADLC,int AIdentifier, string ADatas);</code>
功能介绍	异步发送 CAN 报文
调用位置	设备连接后
输入参数	AldxChn:通道索引 AlsRemote:是否远程帧 AlsExtended:是否扩展帧 ADLC:数据长度 AIdentifier:报文 ID ADatas:报文数据
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.transmit_can_async_verbose(0,false,false,8,0xF, "1,2,3,4,5,6,7,8"); </pre>

52.transmit_can_sync_verbose

函数名称	<code>void transmit_can_sync_verbose(int AldxChn,bool AlsRemote, bool AlsExtended,int ADLC,int AIdentifier, string ADatas,int ATimeoutMs, out bool ASuccess);</code>
功能介绍	同步发送 CAN 报文
调用位置	设备连接后
输入参数	AldxChn:通道索引 AlsRemote:是否远程帧 AlsExtended:是否扩展帧 ADLC:数据长度 AIdentifier:报文 ID ADatas:报文数据 ATimeoutMs:超时时间 ASuccess:报文发送是否成功
返回值	无
示例	<pre> bool ret; TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); </pre>

	<pre>com.transmit_can_sync_verbose(0, false, false, 8, 0x2, "2,3,4,5,6,7,8,9", 1000, out ret);</pre>
--	--

53.transmit_canfd_async_verbose

函数名称	<pre>void transmit_canfd_async_verbose(int AldxChn,bool AlsExtended, bool AlsEDL,bool AlsBRS,bool AlsESI,int ADLC,int AIdentifier, string ADatas);</pre>
功能介绍	异步发送 CANFD 报文
调用位置	设备连接后
输入参数	AldxChn:通道索引 AlsExtended:是否扩展帧 AlsEDL:是否为 CANFD 帧 AlsBRS:是否为可变波特率 AlsESI:错误状态指示 ADLC:数据长度 AIdentifier:数据 ID ADatas:报文数据
返回值	无
示例	<pre>TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.transmit_canfd_async_verbose(0, false, true, false, false, 8, 0x7, "1,2,3,4,5,6,7,8");</pre>

54.transmit_canfd_sync_verbose

函数名称	<pre>void transmit_canfd_sync_verbose(int AldxChn,bool AlsExtended, bool AlsEDL,bool AlsBRS,bool AlsESI,int ADLC,int AIdentifier, string ADatas, int ATimeoutMs, out bool ASuccess);</pre>
功能介绍	同步发送 CANFD 报文
调用位置	设备连接后
输入参数	AldxChn:通道索引 AlsExtended:是否扩展帧 AlsEDL:是否为 CANFD 帧 AlsBRS:是否为可变波特率 AlsESI:错误状态指示 ADLC:数据长度

	AIdentifier:数据 ID ADatas:报文数据 ATimeoutMs:超时时间 ASuccess:报文发送是否成功
返回值	无
示例	<pre>bool ret; TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.transmit_canfd_sync_verbose(0, false, true, false, false, 15, 0x8, "1,2,3,4,5,6,7,8", 1000, out ret);</pre>

55.transmit_lin_async_verbose

函数名称	void transmit_lin_async_verbose(int AldxChn,int ADLC,int AIdentifier, string ADatas);
功能介绍	异步发送 LIN 报文
调用位置	设备连接后
输入参数	AldxChn:通道索引 ADLC:数据长度 AIdentifier:数据 ID ADatas:报文数据
返回值	无
示例	<pre>TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.transmit_lin_async_verbose(0, 8, 0X1, "1,2,3,4,5,6,7,8");</pre>

56.transmit_lin_sync_verbose

函数名称	void transmit_lin_sync_verbose(int AldxChn,int ADLC,int AIdentifier, string ADatas, int ATimeoutMs, out bool ASuccess);
功能介绍	同步发送 LIN 报文
调用位置	设备连接后
输入参数	AldxChn:通道索引 ADLC:数据长度 AIdentifier:数据 ID ADatas:报文数据

	ATimeoutMs:超时时间 ASuccess:报文发送是否成功
返回值	无
示例	<pre>bool ret; TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.transmit_lin_sync_verbose(0, 8, 0x2, "1,2,3,4,5,6,7,8", 1000, out ret);</pre>

57.add_cyclic_msg_can_verbose

函数名称	void add_cyclic_msg_can_verbose(int AldxChn,bool AlsExtended,int AIdentifier,int ADLC,string AData, int APeriodMs);
功能介绍	周期发送 CAN 报文
调用位置	设备连接后
输入参数	AldxChn:通道索引 AlsExtended:是否扩展帧 AIdentifier:数据 ID ADLC:数据长度 AData:报文数据 APeriodMs:周期时间 ms
返回值	无
示例	<pre>TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.add_cyclic_msg_can_verbose(0,false,0x123,8, "0x1,0x2,0x3,0x4,0x5,0x6,0x7,0x8", 100);</pre>

58.add_cyclic_msg_canfd_verbose

函数名称	void add_cyclic_msg_canfd_verbose(int AldxChn,bool AlsExtended, int AIdentifier,int ADLC,string AData, int APeriodMs);
功能介绍	周期发送 CANFD 报文
调用位置	设备连接后
输入参数	AldxChn:通道索引 AlsExtended:是否扩展帧 AIdentifier:数据 ID

	ADLC:数据长度 ADatas:报文数据 APeriodMs:周期时间 ms
返回值	无
示例	<pre> TSApplicationClass app = new TSApplicationClass(); TSCOM com = app.TSCOM(); com.add_cyclic_msg_canfd_verbose(0,false,0x345,8, "0x1,0x2,0x3,0x4,0x5,0x6,0x7,0x8", 100); </pre>

59.delete_cyclic_msg_can_verbose

函数名称	void delete_cyclic_msg_can_verbose(int AldxChn,bool AlsExtended, int AIdentifier);
功能介绍	停止周期发送 CAN 报文
调用位置	设备连接后
输入参数	AldxChn:通道索引 AlsExtended:是否扩展帧 AIdentifier:数据 ID
返回值	无
示例	<pre> TSApplicationClass app = new TSApplicationClass(); TSCOM com = app.TSCOM(); com.delete_cyclic_msg_can_verbose(0,false,0X20); </pre>

60.delete_cyclic_msg_canfd_verbose

函数名称	void delete_cyclic_msg_canfd_verbose(int AldxChn,bool AlsExtended, int AIdentifier);
功能介绍	停止周期发送 CANFD 报文
调用位置	设备连接后
输入参数	AldxChn:通道索引 AlsExtended:是否扩展帧 AIdentifier:数据 ID
返回值	无
示例	<pre> TSApplicationClass app = new TSApplicationClass(); TSCOM com = app.TSCOM(); </pre>

	<code>com.delete_cyclic_msg_canfd_verbose(0,false,0X20);</code>
--	---

61.can_rbs_start

函数名称	<code>void can_rbs_start();</code>
功能介绍	启动 CAN RBS 仿真
调用位置	设备连接后
输入参数	无
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.can_rbs_start(); </pre>

62.can_rbs_stop

函数名称	<code>void can_rbs_stop();</code>
功能介绍	停止 CAN RBS 仿真
调用位置	设备连接后
输入参数	无
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.can_rbs_stop(); </pre>

63.can_rbs_is_running

函数名称	<code>void can_rbs_is_running(out bool AIsRunning);</code>
功能介绍	检查 CAN RBS 是否正在运行

调用位置	
输入参数	AlsRunning:CAN RBS 仿真运行状态
返回值	无
示例	<pre>bool isRunning; TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.can_rbs_is_running(out isRunning);</pre>

64.can_rbs_configure

函数名称	<pre>void can_rbs_configure(bool AAutoStart, bool AAutoSendOnModification, bool AActivateNodeSimulation, TTSRBSInitValueOptions AInitValueOptions);</pre>
功能介绍	CAN RBS 仿真配置
调用位置	
输入参数	<p>AAutoStart:自动启动 RBS 仿真</p> <p>AAutoSendOnModification:信号修改时发送报文</p> <p>AActivateNodeSimulation:是否激活节点行为模拟</p> <p>AInitValueOptions:信号初始值</p>
返回值	无
示例	<pre>TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.can_rbs_configure(false, true, true, TTSRBSInitValueOptions.rivUseDB);</pre>

65.can_rbs_activate_all_networks

函数名称	<pre>void can_rbs_activate_all_networks(bool AEnable, bool AIncludingChildren);</pre>
功能介绍	设置激活或停止所有 CAN 网络
调用位置	
输入参数	<p>AEnable:使能</p> <p>AIncludingChildren:激活包括子节点</p>

返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.can_rbs_activate_all_networks(true, false); </pre>

66.can_rbs_activate_network_by_name

函数名称	<pre> void can_rbs_activate_network_by_name(int AldxChn,bool AEnable, string ANetworkName,bool AIncludingChildren); </pre>
功能介绍	是否激活或停用网络
调用位置	
输入参数	AldxChn:通道序号 AEnable:使能 ANetworkName:网络节点名 AIncludingChildren:激活包括子节点
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.can_rbs_activate_network_by_name(0,true, "CAN_FD_Powertrain",false); </pre>

67.can_rbs_activate_node_by_name

函数名称	<pre> void can_rbs_activate_node_by_name(int AldxChn,bool AEnable, string ANetworkName, string ANodeName,bool AIncludingChildren); </pre>
功能介绍	是否激活或停用节点
调用位置	
输入参数	AldxChn:通道序号 AEnable:使能 ANetworkName:网络节点名 ANodeName:ECU 节点 AIncludingChildren:激活包括子节点
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); </pre>

	<code>com.can_rbs_activate_node_by_name(0, true, "CAN_FD_Powertrain", "Engine", false);</code>
--	--

68.can_rbs_activate_message_by_name

函数名称	<code>void can_rbs_activate_message_by_name(int AldxChn,bool AEnable, string ANetworkName,string ANodeName, string AMsgName);</code>
功能介绍	是否激活或停用报文仿真
调用位置	
输入参数	AldxChn:通道序号 AEnable:使能 ANetworkName:网络节点名 ANodeName:ECU 节点 AMsgName:报文名
返回值	无
示例	<code>TSApplicationClass app = new TSApplicationClass(); TSCOM com = app.TSCOM(); com.can_rbs_activate_message_by_name(0,true,"CAN_FD_Powertrain", "Engine", "EngineData");</code>

69.can_rbs_get_signal_value_by_element

函数名称	<code>void can_rbs_get_signal_value_by_element(int AldxChn, string ANetworkName,string ANodeName, string AMsgName, string ASignalName, out double AValue);</code>
功能介绍	使用元素名称从 CAN RBS 获取信号实时值
调用位置	
输入参数	AldxChn:通道序号 ANetworkName:网络节点名 ANodeName:ECU 节点 AMsgName:报文名 ASignalName:信号名 AValue:信号值
返回值	无
示例	<code>double value; TSApplicationClass app = new TSApplicationClass();</code>

	<pre>TSCOM com = app.TSCOM(); com.can_rbs_get_signal_value_by_element(0,"CAN_FD_Powertrain", "Engine", "EngineData","EngTemp",out value);</pre>
--	---

70.can_rbs_get_signal_value_by_address

函数名称	void can_rbs_get_signal_value_by_address(string ASymbolAddress, out double AValue);
功能介绍	使用信号数据库地址从 CAN RBS 获取信号实时值
调用位置	
输入参数	ASymbolAddress:信号地址 AValue:信号值
返回值	无
示例	<pre>double value; TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.can_rbs_get_signal_value_by_address("0/CAN_FD_Powertrain/Engine/EngineData/EngTemp",out value);</pre>

71.can_rbs_set_signal_value_by_element

函数名称	void can_rbs_set_signal_value_by_element(int AldxChn, string ANetworkName, string ANodeName, string AMsgName, string ASignalName,double AValue);
功能介绍	使用元素名称从 CAN RBS 设置信号实时值
调用位置	
输入参数	AldxChn:通道索引 ANetworkName:网络节点名 ANodeName:ECU 节点名 AMsgName:报文名 ASignalName:信号名 AValue:信号值
返回值	无
示例	<pre>TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.can_rbs_set_signal_value_by_element(0, "CAN_FD_Powertrain",</pre>

	"Engine", "EngineData", "EngTemp",10);
--	--

72.can_rbs_set_signal_value_by_address

函数名称	void can_rbs_set_signal_value_by_address(string ASymbolAddress, out double AValue);
功能介绍	使用信号数据库地址从 CAN RBS 设置信号实时值
调用位置	
输入参数	ASymbolAddress:信号地址 AValue:信号值
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.can_rbs_set_signal_value_by_element("0/CAN_FD_Powertrain/Engine/EngineData/EngTemp",10); </pre>

73.fifo_enable_receive_fifo

函数名称	void fifo_enable_receive_fifo();
功能介绍	启动接收 fifo
调用位置	
输入参数	无
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.fifo_enable_receive_fifo(); </pre>

74.fifo_disable_receive_fifo

函数名称	void fifo_disable_receive_fifo();
功能介绍	关闭接收 fifo

调用位置	
输入参数	无
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.fifo_disable_receive_fifo(); </pre>

75.fifo_enable_receive_error_frames

函数名称	void fifo_enable_receive_error_frames();
功能介绍	启动接收错误帧
调用位置	
输入参数	无
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.fifo_enable_receive_error_frames(); </pre>

76.fifo_disable_receive_error_frames

函数名称	void fifo_disable_receive_error_frames();
功能介绍	关闭接收错误帧
调用位置	
输入参数	无
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.fifo_disable_receive_error_frames(); </pre>

77.fifo_receive_can_msg

函数名称	<code>bool fifo_receive_can_msg(int AldxChnReq,bool AIncludeTx, out int AldxChnResp, out bool AIsRemote, out bool AIsExtended, out int ADLC, out int AIdentifier, out long ATimestampUs, out string AData);</code>
功能介绍	读取 CAN 报文
调用位置	设备连接后
输入参数	AldxChnReq:通道索引 AIncludeTx:包含发送报文 AldxChnResp:通道索引 AIsRemote:、远程帧 AIsExtended:扩展帧 ADLC:数据长度 AIdentifier:报文 ID ATimestampUs:时间戳 AData:报文数据
返回值	函数执行结果 true:成功
示例	<pre>int res, dlc, id; bool remote, extended, edl, rbs; long time; string data; TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); if (com.fifo_receive_can_msg(0, true, out res, out remote, out extended, out dlc, out id, out time, out data)) app.log(id.ToString("X"), 3);</pre>

78.fifo_receive_canfd_msg

函数名称	<code>bool fifo_receive_canfd_msg(int AldxChnReq, bool AIncludeTx, out int AldxChnResp, out bool AIsRemote, out bool AIsExtended, out bool AIsEDL, out bool AIsBRS, out int ADLC, out int AIdentifier, out long ATimestampUs,out string AData);</code>
功能介绍	读取 CANFD 报文
调用位置	设备连接后
输入参数	AldxChnReq:通道索引 AIncludeTx:包含发送报文

	AldxChnResp:报文通道 AlsRemote:、远程帧 AlsExtended:扩展帧 AlsEDL:CANFD 帧 AlsBRS:BRS 模式 ADLC:数据长度 AIdentifier:报文 ID ATimestampUs:时间戳 AData:报文数据
返回值	函数执行结果 true:成功
示例	<pre> int res, dlc, id; bool remote, extended, edl, rbs; long time; string data; TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); if (com.fifo_receive_canfd_msg(0, true, out res, out remote, out extended, out edl, out rbs, out dlc, out id, out time, out data)) app.log(id.ToString("X"), 3); </pre>

79.fifo_receive_lin_msg

函数名称	bool fifo_receive_lin_msg(int AldxChnReq,bool AIncludeTx, out int AldxChnResp, out int ADLC, out int AIdentifier, out long ATimestampUs,out string AData);
功能介绍	读取 LIN 报文
调用位置	设备连接后
输入参数	AldxChnReq:通道索引 AIncludeTx:包含发送报文 AldxChnResp:报文通道 ADLC:数据长度 AIdentifier:报文 ID ATimestampUs:时间戳 AData:报文数据
返回值	函数执行结果 true:成功
示例	<pre> int res, dlc, id; long time; string data; TSAApplicationClass app = new TSAApplicationClass(); </pre>

	<pre>TSCOM com = app.TSCOM(); if (com.fifo_receive_lin_msg(0, true, out res, out dlc, out id, out time, out data)) app.log(id.ToString("X"), 3);</pre>
--	--

80.fifo_receive_fastlin_msg

函数名称	<pre>bool fifo_receive_fastlin_msg(int AldxChnReq,bool AIncludeTx, out int AldxChnResp, out int ADLC, out int AIdentifier, out long ATimestampUs,out string ADatas);</pre>
功能介绍	读取 LIN 报文
调用位置	设备连接后
输入参数	AldxChnReq:通道索引 AIncludeTx:包含发送报文 AldxChnResp:报文通道 ADLC:数据长度 AIdentifier:报文 ID ATimestampUs:时间戳 ADatas:报文数据
返回值	函数执行结果 true:成功
示例	<pre>int res, dlc, id; long time; string data; TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); if (com.fifo_receive_fastlin_msg(0, true, out res, out dlc, out id, out time, out data)) app.log(id.ToString("X"), 3);</pre>

81.fifo_clear_can_receive_buffers

函数名称	<pre>bool fifo_clear_can_receive_buffers(int AldxChn);</pre>
功能介绍	清除 CAN 接收缓存
调用位置	
输入参数	AldxChnReq:通道索引

返回值	函数执行结果 true:成功
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.fifo_clear_can_receive_buffers(0); </pre>

82.fifo_clear_canfd_receive_buffers

函数名称	bool fifo_clear_canfd_receive_buffers(int AldxChn);
功能介绍	清除 CANFD 接收缓存
调用位置	
输入参数	AldxChnReq:通道索引
返回值	函数执行结果 true:成功
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.fifo_clear_canfd_receive_buffers(0); </pre>

83.fifo_clear_lin_receive_buffers

函数名称	bool fifo_clear_lin_receive_buffers(int AldxChn);
功能介绍	清除 LIN 接收缓存
调用位置	
输入参数	AldxChnReq:通道索引
返回值	函数执行结果 true:成功
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.fifo_clear_lin_receive_buffers(0); </pre>

84.fifo_clear_fastlin_receive_buffers

函数名称	bool fifo_clear_fastlin_receive_buffers(int AldxChn);
------	---

功能介绍	清除 fastlin 接收缓存
调用位置	
输入参数	AldxChnReq:通道索引
返回值	函数执行结果 true:成功
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.fifo_clear_fastlin_receive_buffers(0); </pre>

85.fifo_read_can_buffer_frame_count

函数名称	bool fifo_read_can_buffer_frame_count(int AldxChn, out int ACount);
功能介绍	读取 CAN 缓冲区报文帧数
调用位置	
输入参数	AldxChnReq:通道索引 ACount:缓存帧数
返回值	函数执行结果 true:成功
示例	<pre> int count; TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.fifo_read_can_buffer_frame_count(0,out count); </pre>

86.fifo_read_can_tx_buffer_frame_count

函数名称	bool fifo_read_can_tx_buffer_frame_count(int AldxChn, out int ACount);
功能介绍	读取 CAN 缓冲区发送报文帧数
调用位置	
输入参数	AldxChnReq:通道索引 ACount:缓存帧数
返回值	函数执行结果 true:成功

示例	<pre>int count; TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.fifo_read_can_tx_buffer_frame_count(0,out count);</pre>
----	---

87.fifo_read_can_rx_buffer_frame_count

函数名称	bool fifo_read_can_rx_buffer_frame_count(int AldxChn, out int ACount);
功能介绍	读取 CAN 缓冲区接收报文帧数
调用位置	
输入参数	AldxChnReq:通道索引 ACount:缓存帧数
返回值	函数执行结果 true:成功
示例	<pre>int count; TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.fifo_read_can_rx_buffer_frame_count(0,out count);</pre>

88.fifo_read_canfd_buffer_frame_count

函数名称	bool fifo_read_canfd_buffer_frame_count(int AldxChn, out int ACount);
功能介绍	读取 CAN FD 缓冲区报文帧数
调用位置	
输入参数	AldxChnReq:通道索引 ACount:缓存帧数
返回值	函数执行结果 true:成功
示例	<pre>int count; TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.fifo_read_canfd_buffer_frame_count(0,out count);</pre>

89.fifo_read_canfd_tx_buffer_frame_count

函数名称	<code>bool fifo_read_canfd_tx_buffer_frame_count(int AldxChn, out int ACount);</code>
功能介绍	读取 CAN FD 缓冲区发送报文帧数
调用位置	
输入参数	AldxChnReq:通道索引 ACount:缓存帧数
返回值	函数执行结果 true:成功
示例	<pre>int count; TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.fifo_read_canfd_tx_buffer_frame_count(0,out count);</pre>

90.fifo_read_canfd_rx_buffer_frame_count

函数名称	<code>bool fifo_read_canfd_rx_buffer_frame_count(int AldxChn, out int ACount);</code>
功能介绍	读取 CAN FD 缓冲区接收报文帧数
调用位置	
输入参数	AldxChnReq:通道索引 ACount:缓存帧数
返回值	函数执行结果 true:成功
示例	<pre>int count; TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.fifo_read_canfd_rx_buffer_frame_count(0,out count);</pre>

91.fifo_read_lin_buffer_frame_count

函数名称	<code>bool fifo_read_lin_buffer_frame_count(int AldxChn, out int ACount);</code>
功能介绍	读取 LIN 缓冲区报文帧数

调用位置	
输入参数	AldxChnReq:通道索引 ACount:缓存帧数
返回值	函数执行结果 true:成功
示例	<pre>int count; TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.fifo_read_lin_buffer_frame_count(0,out count);</pre>

92.fifo_read_lin_tx_buffer_frame_count

函数名称	bool fifo_read_lin_tx_buffer_frame_count(int AldxChn, out int ACount);
功能介绍	读取 LIN 缓冲区发送报文帧数
调用位置	
输入参数	AldxChnReq:通道索引 ACount:缓存帧数
返回值	函数执行结果 true:成功
示例	<pre>int count; TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.fifo_read_lin_tx_buffer_frame_count(0,out count);</pre>

93.fifo_read_lin_rx_buffer_frame_count

函数名称	bool fifo_read_lin_rx_buffer_frame_count(int AldxChn, out int ACount);
功能介绍	读取 LIN 缓冲区接收报文帧数
调用位置	
输入参数	AldxChnReq:通道索引 ACount:缓存帧数
返回值	函数执行结果 true:成功
示例	<pre>int count; TSAApplicationClass app = new TSAApplicationClass();</pre>

	<pre>TSCOM com = app.TSCOM(); com.fifo_read_lin_rx_buffer_frame_count(0,out count);</pre>
--	---

94.fifo_read_fastlin_buffer_frame_count

函数名称	bool fifo_read_fastlin_buffer_frame_count(int AldxChn, out int ACount);
功能介绍	读取 fastlin 缓冲区报文帧数
调用位置	
输入参数	AldxChnReq:通道索引 ACount:缓存帧数
返回值	函数执行结果 true:成功
示例	<pre>int count; TSApplicationClass app = new TSApplicationClass(); TSCOM com = app.TSCOM(); com.fifo_read_fastlin_buffer_frame_count(0,out count);</pre>

95.fifo_read_fastlin_tx_buffer_frame_count

函数名称	bool fifo_read_fastlin_tx_buffer_frame_count(int AldxChn, out int ACount);
功能介绍	读取 fastlin 缓冲区发送报文帧数
调用位置	
输入参数	AldxChnReq:通道索引 ACount:缓存帧数
返回值	函数执行结果 true:成功
示例	<pre>int count; TSApplicationClass app = new TSApplicationClass(); TSCOM com = app.TSCOM(); com.fifo_read_fastlin_tx_buffer_frame_count(0,out count);</pre>

96.fifo_read_fastlin_rx_buffer_frame_count

函数名称	bool fifo_read_fastlin_rx_buffer_frame_count(int AldxChn, out int ACount);
------	--

功能介绍	读取 fastlin 缓冲区接收报文帧数
调用位置	
输入参数	AIdxChnReq:通道索引 ACount:缓存帧数
返回值	函数执行结果 true:成功
示例	<pre>int count; TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.fifo_read_fastlin_rx_buffer_frame_count(0,out count);</pre>

97.flexray_rbs_start

函数名称	void flexray_rbs_start();
功能介绍	启动 FlexRay RBS 引擎
调用位置	
输入参数	无
返回值	无
示例	<pre>TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.flexray_rbs_start();</pre>

98.flexray_rbs_stop

函数名称	void flexray_rbs_stop();
功能介绍	停止 FlexRay RBS 引擎
调用位置	
输入参数	无
返回值	无

示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.flexray_rbs_stop(); </pre>
----	--

99.flexray_rbs_is_running

函数名称	void flexray_rbs_is_running(out bool AIsRunning);
功能介绍	检查 FlexRay RBS 是否正在运行
调用位置	
输入参数	AIsRunning:FlexRay RBS 仿真运行状态
返回值	无
示例	<pre> bool state = false; TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.flexray_rbs_is_running(out state); if(state) app.log("FlexRay RBS 运行中",3); </pre>

100.flexray_rbs_configure

函数名称	void flexray_rbs_configure(bool AAutoStart, bool AAutoSendOnModification, bool AActivateECUSimulation, TTSRBSInitValueOptions AInitValueOptions);
功能介绍	FlexRay RBS 仿真配置
调用位置	
输入参数	AAutoStart:自动启动 RBS 仿真 AAutoSendOnModification:信号修改时发送报文 AActivateECUSimulation:是否激活 ECU 模拟行为模拟 AInitValueOptions:信号初始值
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.flexray_rbs_configure(true, true, false, </pre>

	TTSRBSInitValueOptions.rivUseDB);
--	-----------------------------------

101.flexray_rbs_activate_all_clusters

函数名称	void flexray_rbs_activate_all_clusters(bool AEnable, bool AIncludingChildren);
功能介绍	设置激活或停用所有 FlexRay 网络
调用位置	
输入参数	AEnable:使能 AIncludingChildren:激活包括子节点
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.flexray_rbs_activate_all_clusters(true, false); </pre>

102.flexray_rbs_activate_cluster_by_name

函数名称	void flexray_rbs_activate_cluster_by_name(int AldxChn,bool AEnable, string AClusterName, bool AIncludingChildren);
功能介绍	是否激活或停用 FlexRay 网络
调用位置	
输入参数	AldxChn:通道索引 AEnable:使能 AClusterName:分组名 AIncludingChildren:激活包括子节点
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.flexray_rbs_activate_cluster_by_name(0, true, "PowerTrain", false); </pre>

103.flexray_rbs_activate_ecu_by_name

函数名称	void flexray_rbs_activate_ecu_by_name(int AldxChn,bool AEnable, string AClusterName,string AECUName,bool AIncludingChildren);
------	---

功能介绍	是否激活或停用 ECU 节点
调用位置	
输入参数	AldxChn:通道索引 AEnable:使能 AClusterName:分组名 AECUName:ECU 名 AIncludingChildren:激活包括子节点
返回值	无
示例	<pre> TSMasterApplicationClass app = new TSMasterApplicationClass(); TSMasterCOM com = app.TSMasterCOM(); com.flexray_rbs_activate_ecu_by_name(0, true, "PowerTrain", "BSC", false); </pre>

104.flexray_rbs_activate_frame_by_name

函数名称	<pre> void flexray_rbs_activate_frame_by_name(int AldxChn,bool AEnable, string AClusterName,string AECUName, string AFrameName); </pre>
功能介绍	是否激活或停用 FlexRay 报文仿真
调用位置	
输入参数	AldxChn:通道索引 AEnable:使能 AClusterName:分组名 AECUName:ECU 名 AFrameName:报文名
返回值	无
示例	<pre> TSMasterApplicationClass app = new TSMasterApplicationClass(); TSMasterCOM com = app.TSMasterCOM(); com.flexray_rbs_activate_frame_by_name(0, true, "PowerTrain", "BSC", "BrakeControl"); </pre>

105.flexray_rbs_get_signal_value_by_element

函数名称	<pre> void flexray_rbs_get_signal_value_by_element(int AldxChn,string AClusterName,string AECUName, string AFrameName, string ASignalName, out double AValue); </pre>
------	--

功能介绍	使用元素名称从 FlexRay RBS 获取信号实时值
调用位置	
输入参数	AldxChn:通道索引 AClusterName:分组名 AECUName:ECU 名 AFrameName:报文名 ASignalName:信号名 AValue:信号值
返回值	无
示例	<pre>double value; TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.flexray_rbs_get_signal_value_by_element(0, "PowerTrain", "BSC", "BrakeControl", "BrakePressure", out value);</pre>

106.flexray_rbs_get_signal_value_by_address

函数名称	void flexray_rbs_get_signal_value_by_address(string ASymbolAddress, out double AValue) ;
功能介绍	使用信号数据库地址从 FlexRay RBS 获取信号实时值
调用位置	
输入参数	ASymbolAddress:信号地址 AValue:信号值
返回值	无
示例	<pre>double value; TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.flexray_rbs_get_signal_value_by_address("0/PowerTrain/BSC/BrakeControl/BrakePressure", out value);</pre>

107.flexray_rbs_set_signal_value_by_element

函数名称	void flexray_rbs_set_signal_value_by_element(int AldxChn, string AClusterName, string AECUName, string AFrameName, string ASignalName, double AValue);
------	--

功能介绍	使用元素名称从 FlexRay RBS 设置信号实时值
调用位置	
输入参数	AIdxChn:通道索引 AClusterName:分组名 AECUName:ECU 名 AFrameName:报文名 ASignalName:信号名 AValue:信号值
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.flexray_rbs_set_signal_value_by_element(0, "PowerTrain", "BSC", "BrakeControl", "BrakePressure",0xff); </pre>

108.flexray_rbs_set_signal_value_by_address

函数名称	void flexray_rbs_set_signal_value_by_address(string ASymbolAddress, double AValue) ;
功能介绍	使用信号数据库地址从 FlexRay RBS 设置信号实时值
调用位置	
输入参数	ASymbolAddress:信号地址 AValue:信号值
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.flexray_rbs_set_signal_value_by_address("0/PowerTrain/BSC/BrakeControl/BrakePressure",0xff); </pre>

109.flexray_rbs_enable

函数名称	void flexray_rbs_enable(bool AEnable) ;
功能介绍	临时启用或禁用 FlexRay RBS 引擎，此功能用于在 RBS 引擎启动之前或之后对其进行后期配置
调用位置	

输入参数	AEnable:使能
返回值	无
示例	<pre>//禁用 rbs_start 之后的消息传输 TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.flexray_rbs_enable(false); com.flexray_rbs_start(); com.flexray_rbs_set_signal_value_by_address("0/PowerTrain/Engine/EngineData/EngForce",1.23); com.flexray_rbs_enable(true); //启用 RBS 后传输消息</pre>

110.flexray_rbs_batch_set_start

函数名称	void flexray_rbs_batch_set_start() ;
功能介绍	开始信号批量设置操作，在此调用后，所有信号设置被缓存，直到 can_rbs_batch_set_end 被调用，这确保当一个帧内的多个信号被设置时，只有一个帧被触发
调用位置	
输入参数	无
返回值	无
示例	<pre>TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.flexray_rbs_batch_set_start(); //在 can_rbs_batch_set_end 之前不会触发消息 com.flexray_rbs_batch_set_signal("0/cluster1/ecu1/frame1/sgn1", 1); com.flexray_rbs_batch_set_signal("0/cluster1/ecu1/frame1/sgn2", 1); com.flexray_rbs_batch_set_end(); //将触发消息 "frame1"</pre>

111.flexray_rbs_batch_set_end

函数名称	void flexray_rbs_batch_set_end() ;
功能介绍	停止信号批量设置操作，在此调用之后，所有缓存的信号都被更新，这确保了当一个帧中的多个信号被设置时，只有一个帧被触发

调用位置	
输入参数	无
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.flexray_rbs_batch_set_start(); //在 can_rbs_batch_set_end 之前不会触发消息 com.flexray_rbs_batch_set_signal("0/cluster1/ecu1/frame1/sgn1", 1); com.flexray_rbs_batch_set_signal("0/cluster1/ecu1/frame1/sgn2", 1); com.flexray_rbs_batch_set_end(); //将触发消息 "frame1" </pre>

112.flexray_rbs_batch_set_signal

函数名称	void flexray_rbs_batch_set_signal(string AAddr, double AValue);
功能介绍	以批量处理模式设置 FlexRay RBS 中的信号
调用位置	
输入参数	AAddr:信号地址 AValue:信号值
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.flexray_rbs_batch_set_start(); //在 can_rbs_batch_set_end 之前不会触发消息 com.flexray_rbs_batch_set_signal("0/cluster1/ecu1/frame1/sgn1", 1); com.flexray_rbs_batch_set_signal("0/cluster1/ecu1/frame1/sgn2", 1); com.flexray_rbs_batch_set_end(); //将触发消息 "frame1" </pre>

113.flexray_rbs_set_frame_direction

函数名称	void flexray_rbs_set_frame_direction(int AldxChn, bool AlsTx, string AClusterName, string AECUName, string AFrameName);
功能介绍	将 ECU 的帧设置为 tx 或 rx 帧

调用位置	
输入参数	AldxChn:通道索引 AIsTx:发送帧还是接收帧 AClusterName:分组名 AECUName:ECU 名 AFrameName:报文名
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); //将 BSC 的所有帧设置为发送帧 com.flexray_rbs_set_frame_direction(0, true, "PowerTrain", "BSC", ""); //将特定帧设置为发送帧 com.flexray_rbs_set_frame_direction(0, true, "PowerTrain", "BSC", "BrakeControl"); </pre>

114.flexray_rbs_set_normal_signal

函数名称	void flexray_rbs_set_normal_signal(string ASymbolAddress);
功能介绍	将信号设置为正常信号
调用位置	
输入参数	ASymbolAddress:信号地址
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.flexray_rbs_set_normal_signal("0/PowerTrain/BSC/BrakeControl/BrakePressure"); </pre>

115.flexray_rbs_set_rc_signal

函数名称	void flexray_rbs_set_rc_signal(string ASymbolAddress);
功能介绍	将信号设置为滚动计数器信号
调用位置	

输入参数	ASymbolAddress:信号地址
返回值	无
示例	<pre> TApplicationClass app = new TApplicationClass(); TSCOM com = app.TSCOM(); com.flexray_rbs_set_rc_signal("0/PowerTrain/BSC/BrakeControl/BrakePressure"); </pre>

116.flexray_rbs_set_rc_signal_with_limit

函数名称	void flexray_rbs_set_rc_signal_with_limit(string ASymbolAddress, int ALowerLimit,int AUpperLimit);
功能介绍	将信号设置为滚动计数器信号
调用位置	
输入参数	ASymbolAddress:信号地址 ALowerLimit:最小值 AUpperLimit:最大值
返回值	无
示例	<pre> TApplicationClass app = new TApplicationClass(); TSCOM com = app.TSCOM(); com.flexray_rbs_set_rc_signal_with_limit("0/PowerTrain/BSC/BrakeControl/BrakePressure",0,14); </pre>

117.flexray_rbs_set_crc_signal

函数名称	void flexray_rbs_set_crc_signal(string ASymbolAddress, string AAlgorithmName, int AldxByteStart, int AByteCount);
功能介绍	将信号设置为 CRC 信号
调用位置	
输入参数	ASymbolAddress:信号地址 AAlgorithmName:小程序库算法函数名 AldxByteStart:保护起始字节 AByteCount:保护的字节数
返回值	无

示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.flexray_rbs_set_crc_signal("0/PowerTrain/Engine/EngineData/EngForce","crc.crc8",0, 7); </pre>
----	--

118.flexray_start_net

函数名称	void flexray_start_net(int AldxChn,int ATimeoutMs);
功能介绍	启用指定的 flexray 框架
调用位置	
输入参数	AldxChn:通道索引 ATimeoutMs:超时时间
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.flexray_start_net(0, 1000); </pre>

119.flexray_stop_net

函数名称	void flexray_stop_net(int AldxChn,int ATimeoutMs);
功能介绍	停止指定的 flexray 框架
调用位置	
输入参数	AldxChn:通道索引 ATimeoutMs:超时时间
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.flexray_stop_net(0, 1000); </pre>

120.transmit_lin_wakeup_async

函数名称	void transmit_lin_wakeup_async(int AldxChn, int AWakeupLength, int AWakeupIntervalTime,int AWakeupTimes);
------	---

功能介绍	发送 LIN 唤醒帧
调用位置	
输入参数	AldxChn:通道索引 AWakeupLength:唤醒长度 AWakeupIntervalTime:唤醒间隔时间 AWakeupTimes:唤醒时间
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.transmit_lin_wakeup_async(0, 5, 150, 10); </pre>

121.transmit_lin_gotosleep_async

函数名称	void transmit_lin_gotosleep_async(int AldxChn);
功能介绍	发送 LIN 休眠帧
调用位置	
输入参数	AldxChn:通道索引
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSCOM com = app.TSCOM(); com.transmit_lin_gotosleep_async(0); </pre>

122.show_main_form

函数名称	void show_main_form();
功能介绍	显示 TSMaster 窗体
调用位置	
输入参数	无
返回值	无

示例	<pre>TSApplicationClass app = new TSApplicationClass(); TSFormManager formManager = app.TSFormManager(); formManager.show_main_form();</pre>
----	--

123.hide_main_form

函数名称	void hide_main_form();
功能介绍	隐藏 TSMaster 窗体
调用位置	
输入参数	无
返回值	无
示例	<pre>TSApplicationClass app = new TSApplicationClass(); TSFormManager formManager = app.TSFormManager(); formManager.hide_main_form();</pre>

124.load_project

函数名称	void load_project(string AFileName,bool ADiscardCurrent);
功能介绍	加载 TSMaster 工程
调用位置	
输入参数	AFileName:工程路径 ADiscardCurrent:是否退出当前工程
返回值	无
示例	<pre>TSApplicationClass app = new TSApplicationClass(); TSFormManager formManager = app.TSFormManager(); formManager.load_project("C:\\Users\\LIN\\Desktop\\1", true);</pre>

125.create_project

函数名称	void create_project(bool ADiscardCurrent);
功能介绍	创建新的 TSMaster 工程

调用位置	
输入参数	ADiscardCurrent:是否退出当前工程
返回值	无
示例	<pre>TSApplicationClass app = new TSApplicationClass(); TSFormManager formManager = app.TSFormManager(); formManager.create_project(true);</pre>

126.save_project

函数名称	void save_project(string AFileName);
功能介绍	保存工程
调用位置	
输入参数	AFileName:保存路径
返回值	无
示例	<pre>TSApplicationClass app = new TSApplicationClass(); TSFormManager formManager = app.TSFormManager(); formManager.save_project("C:/Users/LIN/Desktop/2");</pre>

127.show_tab_by_index

函数名称	void show_tab_by_index(int AIndex);
功能介绍	根据索引切换页面
调用位置	
输入参数	AIndex:页面索引
返回值	无
示例	<pre>TSApplicationClass app = new TSApplicationClass(); TSFormManager formManager = app.TSFormManager(); formManager.show_tab_by_index(2);</pre>

128.show_tab_by_name

函数名称	<code>void show_tab_by_name(string AName);</code>
功能介绍	根据页面名切换页面
调用位置	
输入参数	AName:页面名
返回值	无
示例	<pre>TSApplicationClass app = new TSApplicationClass(); TSFormManager formManager = app.TSFormManager(); formManager.show_tab_by_name("页面 1");</pre>

129.get_mp_list

函数名称	<code>void get_mp_list(out string AList);</code>
功能介绍	获取小程序库列表
调用位置	
输入参数	AList:小程序库列表
返回值	无
示例	<pre>string funList; TSApplicationClass app = new TSApplicationClass(); TSMP mp = app.TSMP(); mp.get_mp_list(out funList); app.log(funList,3);</pre>

130.get_mp_function_list

函数名称	<code>void get_mp_function_list(string AMpName, out string AFuncList);</code>
功能介绍	获取小程序库函数列表

调用位置	
输入参数	AMpName:小程序库 AList:函数列表
返回值	无
示例	<pre>string funList; TSAApplicationClass app = new TSAApplicationClass(); TSMP mp = app.TSMP(); mp.get_mp_function_list("crc.mp", out funList); app.log(funList,3);</pre>

131.get_mp_function_prototype

函数名称	void get_mp_function_prototype(string AMpName, string AFuncName, out string APrototype);
功能介绍	获取小程序库函数原型
调用位置	
输入参数	AMpName:小程序库 AFuncName:函数名 APrototype:函数原型
返回值	无
示例	<pre>string proType; TSAApplicationClass app = new TSAApplicationClass(); TSMP mp = app.TSMP(); mp.get_mp_function_prototype("crc.mp", "CRC8",out proType); app.log(proType,3);</pre>

132.dynamic_invoke

函数名称	void dynamic_invoke(string AMpName, string AFuncName, string AInParameters,out string AOutParameters);
功能介绍	动态调用小程序库函数
调用位置	
输入参数	AMpName:小程序库 AFuncName:函数名 AInParameters:输入参数

	AOutParameters:输出参数
返回值	无
示例	<pre>string outPar; TSApplicationClass app = new TSApplicationClass(); TSMP mp = app.TSMP(); mp.dynamic_invoke("ccode8212.mp", "NewCustom_Function1", "1",out outPar); listView1.Items.Add(outPar);</pre>

133.load_mp

函数名称	void load_mp(string AMpFileName);
功能介绍	加载小程序库
调用位置	
输入参数	AMpFileName:小程序库地址
返回值	无
示例	<pre>TSApplicationClass app = new TSApplicationClass(); TSMP mp = app.TSMP(); mp.load_mp("c:\\tmaster\\tmaster\\bin\\rtluidiagnostics.bpl");</pre>

134.unload_mp

函数名称	void unload_mp(string AMpName);
功能介绍	卸载小程序库
调用位置	
输入参数	AMpName:小程序库名
返回值	无
示例	<pre>TSApplicationClass app = new TSApplicationClass(); TSMP mp = app.TSMP(); mp.unload_mp("rtluidiagnostics.bpl");</pre>

135.unload_all_mps

函数名称	void unload_all_mps();
功能介绍	卸载全部小程序库
调用位置	
输入参数	无
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSMP mp = app.TSMP(); mp.unload_all_mps(); </pre>

136.call_system_api

函数名称	int call_system_api(string AFuncName, object AInArgs, out object AOutArgs);
功能介绍	调用系统 API
调用位置	
输入参数	<p>AFuncName:系统函数名</p> <p>AInArgs:输入参数</p> <p>AOutArgs:输出参数 和输入参数对应,如果传入的参数是指针,通过输出参数可以获得到修改后的值</p>
返回值	执行结果 0:执行成功 其他:错误码
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSMP mp = app.TSMP(); app.connect(); string[] inArgs = new string[2];//输入参数 object temp;//输出参数 inArgs[0] = "Application.Connected"; inArgs[1] = "0"; //app.get_system_var_int32(char* ACompleteName,ps32 AValue) mp.call_system_api("app.get_system_var_int32",inArgs,out temp); string outArgs = (string)((object[])temp)[1]; app.log("连接状态" + outArgs,3); </pre>

137.call_library_api

函数名称	int call_library_api(string AFuncName, object AInArgs, out object AOutArgs);
功能介绍	调用小程序库函数
调用位置	
输入参数	AFuncName:小程序库函数名 AInArgs:输入参数 AOutArgs:输出参数 和输入参数对应,如果传入的参数是指针,通过输出参数可以获得到修改后的值
返回值	执行结果 0:执行成功 其他:错误码
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSMP mp = app.TSMP(); string[] inArgs = new string[3];//输入参数 object temp;//输出参数 inArgs[0] = "1"; inArgs[1] = "2"; inArgs[2] = "0"; //s32 Add(s32 A1, s32 A2,s32* A3) int value = mp.call_library_api("ccode3843.Add",inArgs,out temp); app.log((string)((object[])temp)[0], 3); app.log((string)((object[])temp)[1], 3); app.log((string)((object[])temp)[2], 3); </pre>

138.add_online_replay_config

函数名称	void add_online_replay_config(string AFileName, out int AIndex);
功能介绍	添加在线回放引擎
调用位置	TSMaster 打开后
输入参数	AFileName:回放文件名 AIndex:在线回放引擎索引
返回值	无
示例	<pre> int index; TSAApplicationClass app = new TSAApplicationClass(); TSLog log = app.TSLog(); </pre>

	log.add_online_replay_config("C:\\放电.blf",out index);
--	---

139.set_online_replay_config

函数名称	void set_online_replay_config(int AIndex, string AName, string AFileName,bool AAutoStart, bool AIsRepetitiveMode, TTSOnlineReplayTimingMode AStartTimingMode, int AStartDelayTimeMs,bool ASendTx,bool ASendRx, string AMappings);
功能介绍	设置在线回放引擎
调用位置	TSMaster 打开后
输入参数	AIndex:在线回放引擎索引 AName:显示名 AFileName:文件名 AAutoStart:自启动激活 AIsRepetitiveMode:循环模式 AStartTimingMode:启动时序模式 AStartDelayTimeMs:启动延时时间 ASendTx:回放发送报文 ASendRx:回放接收报文 AMappings:回放通道
返回值	无
示例	<pre> TSMasterApplicationClass app = new TSMasterApplicationClass(); TSMasterLog log = app.TSMasterLog(); log.set_online_replay_config(0, "test", "C:\\放电.blf", false, false, TTSOnlineReplayTimingMode.ortImmediately, 0, true, false, "0,1"); </pre>

140.get_online_replay_count

函数名称	void get_online_replay_count(out int ACount);
功能介绍	获取在线回放引擎数
调用位置	
输入参数	ACount:在线回放引擎数

返回值	无
示例	<pre>int count; TSApplicationClass app = new TSApplicationClass(); TSLog log = app.TSLog(); log.get_online_replay_count(out count);</pre>

141.get_online_replay_config

函数名称	<pre>void get_online_replay_config(int AIndex, out string AName, out string AFileName, out bool AAutoStart, out bool AIsRepetitiveMode, out TTSONlineReplayTimingMode AStartTimeMode, out int AStartDelayTimeMs, out bool ASendTx, out bool ASendRx, out string AMappings);</pre>
功能介绍	获取在线回放引擎配置
调用位置	
输入参数	<p>AIndex:在线回放引擎索引 AName:显示名 AFileName:文件名 AAutoStart:自启动激活 AIsRepetitiveMode:循环模式 AStartTimeMode:启动时序模式 AStartDelayTimeMs:启动延时时间 ASendTx:回放发送报文 ASendRx:回放接收报文 AMappings:回放通道</p>
返回值	无
示例	<pre>string name, fileName,mapping; bool autoStart, isRepetitiveMode,seedTX,seedRX; TTSONlineReplayTimingMode startTimeMode; int startDelayTime; TSApplicationClass app = new TSApplicationClass(); TSLog log = app.TSLog(); log.get_online_replay_config(0,out name,out fileName, out autoStart,out isRepetitiveMode,out startTimeMode, out startDelayTime, out seedTX,out seedRX, out mapping);</pre>

142.del_online_replay_config

函数名称	void del_online_replay_config(int AIndex);
功能介绍	按索引删除在线回放引擎
调用位置	
输入参数	AIndex:在线回放引擎索引
返回值	无
示例	<pre>TSApplicationClass app = new TSApplicationClass(); TSLog log = app.TSLog(); log.del_online_replay_config(0);</pre>

143.del_online_replay_configs

函数名称	void del_online_replay_configs();
功能介绍	删除全部在线回放引擎
调用位置	
输入参数	无
返回值	无
示例	<pre>TSApplicationClass app = new TSApplicationClass(); TSLog log = app.TSLog(); log.del_online_replay_configs();</pre>

144.start_online_replay

函数名称	void start_online_replay(int AIndex);
功能介绍	按索引启动在线回放引擎
调用位置	设备连接后

输入参数	AIndex:在线回放引擎索引
返回值	无
示例	<pre> TSApplicationClass app = new TSApplicationClass(); TSLog log = app.TSLog(); log.start_online_replay(0); </pre>

145.start_online_replays

函数名称	void start_online_replays();
功能介绍	启动所有在线回放引擎
调用位置	设备连接后
输入参数	无
返回值	无
示例	<pre> TSApplicationClass app = new TSApplicationClass(); TSLog log = app.TSLog(); log.start_online_replays(); </pre>

146.pause_online_replay

函数名称	void pause_online_replay(int AIndex);
功能介绍	按索引暂停在线回放引擎
调用位置	
输入参数	AIndex:在线回放引擎索引
返回值	无
示例	<pre> TSApplicationClass app = new TSApplicationClass(); TSLog log = app.TSLog(); log.pause_online_replay(0); </pre>

147.pause_online_replays

函数名称	<code>void pause_online_replays();</code>
功能介绍	暂停所有在线回放引擎
调用位置	
输入参数	无
返回值	无
示例	<pre>TSAApplicationClass app = new TSAApplicationClass(); TSLog log = app.TSLog(); log.pause_online_replays();</pre>

148.stop_online_replay

函数名称	<code>void stop_online_replay(int AIndex);</code>
功能介绍	按索引停止在线回放引擎
调用位置	
输入参数	AIndex:在线回放引擎索引
返回值	无
示例	<pre>TSAApplicationClass app = new TSAApplicationClass(); TSLog log = app.TSLog(); log.stop_online_replay(0);</pre>

149.stop_online_replays

函数名称	<code>void stop_online_replays();</code>
功能介绍	停止所有在线回放引擎
调用位置	

输入参数	无
返回值	无
示例	<pre> TSApplicationClass app = new TSApplicationClass(); TSLog log = app.TSLog(); log.stop_online_replays(); </pre>

150.get_online_replay_status

函数名称	<pre> void get_online_replay_status(int AIndex, out TTSONlineReplayStatus AStatus, out float AProgressPercent100); </pre>
功能介绍	获取在线回放引擎状态
调用位置	
输入参数	AIndex:在线回放引擎索引 AStatus:回放状态 AProgressPercent100:回放百分比
返回值	无
示例	<pre> TTSONlineReplayStatus status; float Progress; TSApplicationClass app = new TSApplicationClass(); TSLog log = app.TSLog(); log.get_online_replay_status(0,out status,out Progress); app.log("回放状态:" + status.ToString() + "" + "回放进度:" + Progress.ToString(),3); </pre>

151.load_can_db

函数名称	<pre> void load_can_db(string ADBCFile, string ASupportedChannels, out int Aid); </pre>
功能介绍	加载 CAN 数据库
调用位置	
输入参数	ADBCFile:DBC 路径 ASupportedChannels:加载通道 Aid:数据库 ID
返回值	无

示例	<pre>int dbclId; TSApplicationClass app = new TSApplicationClass(); TSDB db = app.TSDB(); db.load_can_db("C:\\TSMaster\\bin\\Data\\Demo\\Databases \\CAN_FD_Powertrain.dbc" ,"0,1",out dbclId);</pre>
----	--

152.unload_can_db

函数名称	void unload_can_db(int Aid);
功能介绍	按数据库 ID 卸载 CAN 数据库
调用位置	
输入参数	Aid:数据库 ID
返回值	无
示例	<pre>TSApplicationClass app = new TSApplicationClass(); TSDB db = app.TSDB(); db.unload_can_db(0);</pre>

153.unload_can_dbs

函数名称	void unload_can_dbs();
功能介绍	卸载所有 CAN 数据库
调用位置	
输入参数	无
返回值	无
示例	<pre>TSApplicationClass app = new TSApplicationClass(); TSDB db = app.TSDB(); db.unload_can_dbs();</pre>

154.get_can_db_count

函数名称	void get_can_db_count(out int ACount);
功能介绍	获取加载的 CAN 数据库计数
调用位置	
输入参数	ACount:加载的 CAN 数据计数
返回值	无
示例	<pre>int count; TSApplicationClass app = new TSApplicationClass(); TSDB db = app.TSDB(); db.get_can_db_count(out count);</pre>

155.get_can_db_id

函数名称	void get_can_db_id(int AIndex, out int AId);
功能介绍	按索引获取 CAN 数据库 ID
调用位置	
输入参数	AIndex:数据库索引 AId:数据库 ID
返回值	无
示例	<pre>int dbId; TSApplicationClass app = new TSApplicationClass(); TSDB db = app.TSDB(); db.get_can_db_id(0, out dbId);</pre>

156.get_can_db_info

函数名称	int get_can_db_info(int ADatabaseId,int AType, int AIndex, int ASubIndex,out string AValue);
功能介绍	获取数据库信息

调用位置	加载 CAN 数据库后
输入参数	ADatabaseId:数据库 ID AType:信息类型 AIndex:数据库索引 ASubIndex:子索引 AValue:数据库信息
返回值	执行结果 0:成功
示例	<pre>string value; TSAApplicationClass app = new TSAApplicationClass(); TSDB db = app.TSDB(); db.get_can_db_info(0,0,0,0,out value);</pre>

157.set_signal_value_can

函数名称	void set_signal_value_can(ref TCAN AInCAN, string AMsgName, string ASgnName,double AValue, out Array ADatas);
功能介绍	设置信号值
调用位置	加载 CAN 数据库后
输入参数	AInCAN:报文结构体 AMsgName:报文名 ASgnName:信号名 AValue:信号值 ADatas:返回数据
返回值	无
示例	

158.get_signal_value_can

函数名称	void get_signal_value_can(ref TCAN AInCAN, string AMsgName, string ASgnName, out double AValue);
功能介绍	传入的 CAN 报文结构体,获取对应的信号值
调用位置	加载 CAN 数据库后
输入参数	AInCAN:报文结构体 AMsgName:报文名

	ASgnName:信号名 AValue:信号值
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSDB db = app.TSDB(); TCAN tcan = new TCAN(); sbyte[] data = new sbyte[8]; tcan.FDdatas = data; double value; db.get_signal_value_can(ref tcan, "EngineData", "EngTemp", out value); </pre>

159.set_signal_value_canfd

函数名称	void set_signal_value_canfd(ref TCANFD AInCANFD, string AMsgName,string ASgnName,double AValue, out Array ADatas);
功能介绍	设置信号值
调用位置	加载 CAN 数据库后
输入参数	AInCANFD:报文结构体 AMsgName:报文名 ASgnName:信号名 AValue:信号值 ADatas:返回数据
返回值	无
示例	

160.get_signal_value_canfd

函数名称	void get_signal_value_canfd(ref TCANFD AInCANFD, string AMsgName,string ASgnName, out double AValue);
功能介绍	获取信号值
调用位置	加载 CAN 数据库后
输入参数	AInCANFD:报文结构体 AMsgName:报文名

	ASgnName:信号名 AValue:信号值
返回值	无
示例	<pre> TSGlobalApplicationClass app = new TSGlobalApplicationClass(); TSGlobalTSDB db = app.TSGlobalTSDB(); TSGlobalTCANFD tcan = new TSGlobalTCANFD(); sbyte[] data = new sbyte[64]; tcan.FDdatas = data; double value; db.get_signal_value_canfd(ref tcan, "EngineData", "EngTemp", out value); </pre>

161.set_signal_value_can_verbose

函数名称	void set_signal_value_can_verbose(int AldxChn, string AInDdatas, string AMsgName,string ASgnName,double AValue, out string AOutDdatas);
功能介绍	传入的 CAN 报文数据,其中对应的信号位置改为对应的值然后返回数据
调用位置	加载 CAN 数据库后
输入参数	AldxChn:通道索引 AInDdatas:输入数据 AMsgName:报文名 ASgnName:信号名 AValue:信号值 AOutDdatas:返回数据
返回值	无
示例	<pre> string value; TSGlobalApplicationClass app = new TSGlobalApplicationClass(); TSGlobalTSDB db = app.TSGlobalTSDB(); db.set_signal_value_can_verbose(0,"0,0,0,0,0,0,0,0", "EngineData", "EngTemp", 10,out value); app.log(value,3); </pre>

162.get_signal_value_can_verbose

函数名称	void get_signal_value_can_verbose(int AldxChn, string AInDdatas, string AMsgName,string ASgnName, out double AValue);
------	---

功能介绍	传入的 CAN 报文数据,获取对应的信号值
调用位置	加载 CAN 数据库后
输入参数	AIdxChn:通道索引 AInDatas:输入数据 AMsgName:报文名 ASgnName:信号名 AValue:信号值
返回值	无
示例	<pre>double value; TSApplicationClass app = new TSApplicationClass(); TSDB db = app.TSDB(); db.get_signal_value_can_verbose(1, "0xFF,0xFF,0xFF,0xFF,0xFF", "EngineData", "EngTemp", out value); app.log(value.ToString(), 3);</pre>

163.set_signal_value_canfd_verbose

函数名称	<pre>void set_signal_value_canfd_verbose(int AIdxChn, string AInDatas, string AMsgName,string ASgnName,double AValue, out string AOutDatas);</pre>
功能介绍	传入的 CANFD 报文数据,其中对应的信号位置改为对应的值然后返回数据
调用位置	加载 CAN 数据库后
输入参数	AIdxChn:通道索引 AInDatas:输入数据 AMsgName:报文名 ASgnName:信号名 AValue:信号值 AOutDatas 输出数据
返回值	无
示例	<pre>string value; TSApplicationClass app = new TSApplicationClass(); TSDB db = app.TSDB(); db.set_signal_value_canfd_verbose(0,"0,0,0,0,0,0,0", "EngineData", "EngTemp", 10,out value); app.log(value,3);</pre>

164.get_signal_value_canfd_verbose

函数名称	<code>void get_signal_value_canfd_verbose(int AldxChn, string AInDatas, string AMsgName, string ASgnName, out double AValue);</code>
功能介绍	传入的 CAN 报文数据,获取对应的信号值
调用位置	加载 CAN 数据库后
输入参数	AldxChn:通道索引 AInDatas:输入数据 AMsgName:报文名 ASgnName:信号名 AValue:信号值
返回值	无
示例	<pre>double value; TSApplicationClass app = new TSApplicationClass(); TSDB db = app.TSDB(); db.get_signal_value_canfd_verbose(1, "0xFF,0xFF,0xFF,0xFF,0xFF", "EngineData", "EngTemp", out value); app.log(value.ToString(), 3);</pre>

165.load_flexray_db

函数名称	<code>void load_flexray_db(string AFRFile, string ASupportedChannels, out int Aid);</code>
功能介绍	加载 FlexRay 数据库
调用位置	
输入参数	AFRFile:数据库路径 ASupportedChannels:加载通道 Aid:数据库 ID
返回值	无
示例	<pre>int frId; TSApplicationClass app = new TSApplicationClass(); TSDB db = app.TSDB(); db.load_flexray_db("C:\\PowerTrain_v2.xml", "0,1", out frId);</pre>

166.unload_flexray_db

函数名称	<code>void unload_flexray_db(int AId);</code>
功能介绍	按 ID 卸载 FlexRay 数据库
调用位置	
输入参数	AId:数据库 ID
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSDB db = app.TSDB(); db.unload_flexray_db(0); </pre>

167.unload_flexray_dbs

函数名称	<code>void unload_flexray_dbs();</code>
功能介绍	卸载所有 FlexRay 数据库
调用位置	
输入参数	无
返回值	无
示例	<pre> TSAApplicationClass app = new TSAApplicationClass(); TSDB db = app.TSDB(); db.unload_flexray_dbs(); </pre>

168.get_flexray_db_count

函数名称	<code>void get_flexray_db_count(out int ACount);</code>
功能介绍	获取加载的 FlexRay 数据库计数
调用位置	

输入参数	ACount:FlexRay 数据库计数
返回值	无
示例	<pre>int count; TSApplicationClass app = new TSApplicationClass(); TSDB db = app.TSDB(); db.get_flexray_db_count(out count);</pre>

169.get_flexray_db_properties_by_address

函数名称	<pre>void get_flexray_db_properties_by_address(string AAddr, out int ADBIndex, out int ASignalCount, out int AFrameCount, out int AECUCount, out long ASupportedChannelMask, out string AName,out string AComment);</pre>
功能介绍	按地址获取 FlexRay 数据库信息
调用位置	
输入参数	<p>AAddr:数据库地址 ADBIndex:数据库索引 ASignalCount:信号数量 AFrameCount:报文数量 AECUCount:ECU 数量 ASupportedChannelMask:加载数据库的通道掩码 //1011 表示通道 124 都加载了数据库 AName:数据库名 AComment:注释</p>
返回值	无
示例	<pre>int index, sigalCount, frameCount, ecuCount; long channel; string name, comment; TSApplicationClass app = new TSApplicationClass(); TSDB db = app.TSDB(); db.get_flexray_db_properties_by_address("0/PowerTrain", out index, out sigalCount, out frameCount, out ecuCount, out channel, out name, out comment);</pre>

170.get_flexray_db_properties_by_address

函数名称	void get_flexray_db_properties_by_index(int ADBIndex, out int ASignalCount, out int AFrameCount, out int AECUCount, out long ASupportedChannelMask, out string AName, out string AComment);
功能介绍	按索引获取 FlexRay 数据库信息
调用位置	
输入参数	ADBIndex:数据库索引 ASignalCount:信号数量 AFrameCount:报文数量 AECUCount:ECU 数量 ASupportedChannelMask:加载数据库的通道掩码 //1011 表示通道 124 都加载了数据库 AName:数据库名 AComment:注释
返回值	无
示例	<pre>int sigalCount, frameCount, ecuCount; long channel; string name, comment; TSApplicationClass app = new TSApplicationClass(); TSDB db = app.TSDB(); db.get_flexray_db_properties_by_index(0, out sigalCount, out frameCount, out ecuCount, out channel, out name, out comment);</pre>

171.get_flexray_ecu_properties_by_address

函数名称	void get_flexray_ecu_properties_by_address(string AAddr, out int ADBIndex, out int AECUIndex, out int ATxFrameCount, out int ARxFrameCount, out string AName, out string AComment);
功能介绍	按地址获取 ECU 信息
调用位置	
输入参数	AAddr:ECU 地址 ADBIndex:数据库索引 AECUIndex:ECU 索引 ATxFrameCount:发送报文数量 ARxFrameCount:接收报文数量

	AName:ECU 名 AComment:注释
返回值	无
示例	<pre>int ecuIndex, txFromCount, rxFromCount,index; string name, comment; TSApplicationClass app = new TSApplicationClass(); TSDB db = app.TSDB(); db.get_flexray_ecu_properties_by_address("0/PowerTrain/BLU", out index, out ecuIndex, out txFromCount, out rxFromCount, out name, out comment);</pre>

172.get_flexray_ecu_properties_by_index

函数名称	<pre>void get_flexray_ecu_properties_by_index(int ADBIndex, int AECUIndex, out int ATxFrameCount, out int ARxFrameCount, out string AName,out string AComment);</pre>
功能介绍	按索引获取 ECU 信息
调用位置	
输入参数	ADBIndex:数据库索引 AECUIndex:ECU 索引 ATxFrameCount:发送报文数量 ARxFrameCount:接收报文数量 AName:ECU 名 AComment:注释
返回值	无
示例	<pre>int ecuIndex, txFromCount, rxFromCount, index; string name, comment; TSApplicationClass app = new TSApplicationClass(); TSDB db = app.TSDB(); db.get_flexray_ecu_properties_by_index(0, 0,out txFromCount, out rxFromCount, out name, out comment);</pre>

173.get_flexray_frame_properties_by_address

函数名称	<pre>void get_flexray_frame_properties_by_address(string AAddr, out int ADBIndex, out int AECUIndex, out int AFrameIndex, out bool AIsTx, out int AFRChannelMask, out int AFRBaseCycle,</pre>
------	---

	out int AFRCycleRepetition, out bool AFRIStartupFrame, out int AFRSlotId, out long AFRCycleMask, out int ASignalCount, out int AFRDLC, out string AName,out string AComment);
功能介绍	按地址获取报文信息
调用位置	
输入参数	AAddr:报文地址 ADBIndex:数据库索引 AECUIndex:ECU 索引 AFrameIndex:报文索引 AIsTx:报文方向 AFRChannelMask:FR 通道掩码 AB 通道 AFRBaseCycle:BaseCycle AFRCycleRepetition:循环重复 AFRIStartupFrame:启动帧 AFRSlotId:SlotID AFRCycleMask:周期掩码 ASignalCount:信号数量 AFRDLC:数据长度 AName:报文明 AComment:注释
返回值	无
示例	<pre> int frameIndex, frChannel, baseCycle, cycleRepetition, slotId, signalCount, dlc; int index, ecuIndex, sigalCount, frameCount, ecuCount; bool isTx, isStartupFrame; long cycleMask; string name, comment; TSApplicationClass app = new TSApplicationClass(); TSDB db = app.TSDB(); db.get_flexray_frame_properties_by_address("0/PowerTrain/BSC/BrakeControl", out index, out ecuIndex, out frameIndex, out isTx, out frChannel, out baseCycle, out cycleRepetition, out isStartupFrame, out slotId, out cycleMask, out sigalCount, out dlc, out name, out comment); </pre>

174.get_flexray_frame_properties_by_index

函数名称	void get_flexray_frame_properties_by_index(int ADBIndex, int AECUIndex, int AFrameIndex,bool AIsTx, out int AFRChannelMask, out int AFRBaseCycle, out int AFRCycleRepetition,
------	---

	out bool AFRIsStartupFrame, out int AFRSlotId, out long AFRCycleMask, out int ASignalCount, out int AFRDLC, out string AName, out string AComment);
功能介绍	按索引获取报文信息
调用位置	
输入参数	ADBIndex:数据库索引 AECUIndex:ECU 索引 AFrameIndex:报文索引 AIsTx:报文方向 AFRCycleMask:FR 通道掩码 AB 通道 AFRBaseCycle:BaseCycle AFRCycleRepetition:循环重复 AFRIsStartupFrame:启动帧 AFRSlotId:SlotID AFRCycleMask:周期掩码 ASignalCount:信号数量 AFRDLC:数据长度 AName:报文名 AComment:注释
返回值	无
示例	<pre> int frameIndex, frChannel, baseCycle, cycleRepetition, slotId, signalCount, dlc; int sigalCount; bool isStartupFrame; long cycleMask; string name, comment; TSApplicationClass app = new TSApplicationClass(); TSDB db = app.TSDB(); db.get_flexray_frame_properties_by_index(0, 0,0, true, out frChannel, out baseCycle, out cycleRepetition,out isStartupFrame, out slotId, out cycleMask, out sigalCount,out dlc, out name, out comment); </pre>

175.get_flexray_signal_properties_by_address

函数名称	void get_flexray_signal_properties_by_address(string AAddr, out int ADBIndex, out int AECUIndex, out int AFrameIndex, out int ASignalIndex, out bool AIsTx, out TTSSignalType ASignalType, out TTSFlexRayCompuMethod ACompuMethod, out bool AIsIntel, out int AStartBit, out int AUpdateBit, out int ALength, out double AFactor, out double AOffset, out double AInitValue,
------	--

	out string AName, out string AComment);
功能介绍	按地址获取信号信息
调用位置	
输入参数	AAddr: 信号地址 ADBIndex: 数据库索引 AECUIndex: ECU 索引 AFrameIndex: 报文索引 ASignalIndex: 信号索引 AlsTx: 报文方向 ASignalType: 信号类型 ACompuMethod: 计算方法 AlsIntel: 字节序 true:英特尔 false:摩托罗拉 AStartBit: 起始位 AUpdateBit: 更新位 ALength: 长度 AFactor: 放大因子 AOffset: 偏移量 AlnitValue: 初始值 AName: 信号名 AComment: 注释
返回值	无
示例	<pre> int signalIndex, startBit, updateBit, length; int ecuIndex, frameIndex, txFromCount, rxFromCount; int index, sigalCount, frameCount, ecuCount; TTSSignalType signalType; TTSTFlexRayCompuMethod compuMethod; bool isIntel, isTx; double factor, offset, initValue; string name, comment; TSApplicationClass app = new TSApplicationClass(); TSDB db = app.TSDB(); db.get_flexray_signal_properties_by_address("0/PowerTrain/BSC/BrakeControl/BrakePressure", out index, out ecuIndex, out frameIndex, out signalIndex, out isTx, out signalType, out compuMethod, out isIntel, out startBit, out updateBit, out length, out factor, out offset, out initValue, out name, out comment); </pre>

176.get_flexray_signal_properties_by_index

函数名称	void get_flexray_signal_properties_by_index(int ADBIndex, int AECUIndex, int AFrameIndex,int ASignalIndex, bool AIsTx, out TTSSignalType ASignalType, out TTSTFlexRayCompuMethod ACompuMethod, out bool AIsIntel, out int AStartBit, out int AUpdateBit, out int ALength, out double AFactor, out double AOffset, out double AInitValue, out string AName, out string AComment);
功能介绍	按索引获取信号信息
调用位置	
输入参数	<p>ADBIndex:数据库索引 AECUIndex:ECU 索引 AFrameIndex:报文索引 ASignalIndex:信号索引 AIsTx:报文方向 ASignalType:信号类型 ACompuMethod:计算方法 AIsIntel:字节序 true:英特尔 false:摩托罗拉 AStartBit:起始位 AUpdateBit:更新位 ALength:长度 AFactor:放大因子 AOffset:偏移量 AInitValue:初始值 AName:信号名 AComment:注释</p>
返回值	无
示例	<pre>int startBit, updateBit, length; int frameIndex, txFromCount, rxFromCount; int sigalCount, frameCount, ecuCount; TTSSignalType signalType; TTSTFlexRayCompuMethod compuMethod; bool isIntel; double factor, offset, initValue; string name, comment; TSApplicationClass app = new TSApplicationClass(); TSDB db = app.TSDB(); db.get_flexray_signal_properties_by_index(0, 0, 0, 0, true, out signalType, out compuMethod,</pre>

	out isIntel, out startBit, out updateBit, out length, out factor, out offset, out initValue, out name, out comment);
--	---

177.get_flexray_db_id

函数名称	void get_flexray_db_id(int AIndex,out int AId);
功能介绍	按索引获取 FlexRay 数据库 ID
调用位置	
输入参数	AIndex:数据库索引 AId:数据库 ID
返回值	无
示例	<pre>int frId; TSAApplicationClass app = new TSAApplicationClass(); TSDB db = app.TSDB(); db.get_flexray_db_id(0,out frId);</pre>

178.diag_can_create

函数名称	bool diag_can_create(out int pDiagModuleIndex, int AChnIndex, bool ASupportFDCAN,int AMaxDLC,int ARequestID, bool ARequestIDIsStd, int AResponseID, bool AResponseIDIsStd, int AFunctionID,bool AFunctionIDIsStd);
功能介绍	创建诊断模块
调用位置	
输入参数	<p>pDiagModuleIndex:诊断模块索引</p> <p>AChnIndex:CAN 通道</p> <p>ASupportFDCAN:支持 CANFD</p> <p>AMaxDLC:最大 DLC</p> <p>ARequestID:请求 ID</p> <p>ARequestIDIsStd:请求 ID 是否为扩展帧</p> <p>AResponseID:应答 ID</p> <p>AResponseIDIsStd:应答 ID 是否为扩展帧</p> <p>AFunctionID:功能 ID</p> <p>AFunctionIDIsStd:功能 ID 是否为扩展帧</p>
返回值	执行结果

示例	<pre>int diagnosticIndex; TSAApplicationClass app = new TSAApplicationClass(); TSDiagnostic diagnostic = app.TSDiagnostic(); diagnostic.diag_can_create(out diagnosticIndex,0,false,8,0x7c0,false, 0x7c8,false,0x7df,false);</pre>
----	--

179.diag_set_fdmode

函数名称	bool diag_set_fdmode(int ADiagModuleIndex, bool AFDMODE, bool ASupportBRS,int AMaxDLC);
功能介绍	诊断模块使能 CANFD 模式
调用位置	诊断模块创建后
输入参数	ADiagModuleIndex:诊断模块索引 AFDMODE:FD 模式 ASupportBRS:BRS 模式 AMaxDLC:最大 DLC
返回值	执行结果
示例	<pre>int diagnosticIndex; TSAApplicationClass app = new TSAApplicationClass(); TSDiagnostic diagnostic = app.TSDiagnostic(); diagnostic.diag_can_create(out diagnosticIndex, 0, false, 8, 0x7c0, false,0x7c8, false, 0x7df, false); diagnostic.diag_set_fdmode(diagnosticIndex, true, true, 15);</pre>

180.diag_can_delete

函数名称	bool diag_can_delete(int ADiagModuleIndex);
功能介绍	按索引删除诊断模块
调用位置	
输入参数	ADiagModuleIndex:诊断模块索引
返回值	执行结果
示例	<pre>TSAApplicationClass app = new TSAApplicationClass(); TSDiagnostic diagnostic = app.TSDiagnostic(); diagnostic.diag_can_delete(0);</pre>

181.diag_can_delete_all

函数名称	void diag_can_delete_all();
功能介绍	删除所有诊断模块
调用位置	
输入参数	无
返回值	无
示例	<pre>TSAApplicationClass app = new TSAApplicationClass(); TSDiagnostic diagnostic = app.TSDiagnostic(); diagnostic.diag_can_delete_all();</pre>

182.tp_can_request_and_get_response

函数名称	bool tp_can_request_and_get_response(int ADiagModuleIndex, string ARequest, out string AResponse);
功能介绍	发送诊断请求并获取应答
调用位置	诊断模块创建后
输入参数	ADiagModuleIndex:诊断模块索引 ARequest:请求 AResponse:应答
返回值	执行结果
示例	

183.tp_can_request_and_get_response_functional

函数名称	bool tp_can_request_and_get_response_functional(int ADiagModuleIndex,string ARequest, out string AResponse);
功能介绍	发送诊断请求并获取应答
调用位置	诊断模块创建后

输入参数	ADiagModuleIndex:诊断模块索引 ARequest:请求 AResponse:应答
返回值	执行结果
示例	