

TSMaster SDK

TSMaster SDK	1
TSMaster API Introduction.....	3
TSMaster API Documentation.....	4
1.1 Start Using TSMaster API.....	4
1.2 TSMaster API General Information.....	4
1.3 TSMaster API basic data types	9
1.4 TSMaster API Description.....	9
1.4.1 initialize_lib_tsmaster	9
1.4.2 finalize_lib_tsmaster	10
1.4.3 tsapp_set_current_application	10
1.4.4 tsapp_get_current_application	10
1.4.5 tsapp_del_application	10
1.4.6 tsapp_add_application	10
1.4.7 tsapp_set_logger	10
1.4.8 tsapp_get_application_list	10
1.4.9 tsapp_set_can_channel_count	10
1.4.10 tsapp_set_lin_channel_count	11
1.4.11 tsapp_get_can_channel_count	11
1.4.12 tsapp_get_lin_channel_count	11
1.4.13 tsapp_set_mapping	11
1.4.14 tsapp_get_mapping	11
1.4.15 tsapp_del_mapping	11
1.4.16 tsapp_connect	11
1.4.17 tsapp_disconnect	12
1.4.18 tsapp_set_turbo_mode	12
1.4.19 tsapp_get_turbo_mode	12
1.4.20 tsapp_get_error_description	12
1.4.21 tsapp_show_channel_mapping_window	12
1.4.22 tsapp_show_hardware_configuration_window	12
1.4.23 tsapp_show_tsmaster_window	12
1.4.24 tsapp_get_timestamp	13
1.4.25 tsapp_transmit_can_async	13
1.4.26 tsapp_transmit_canfd_async	13

1.4.27	tsapp_transmit_lin_async	13
1.4.28	tsapp_transmit_fastlin_async	13
1.4.29	tsapp_transmit_can_sync	13
1.4.30	tsapp_transmit_canfd_sync	13
1.4.31	tsapp_transmit_lin_sync	13
1.4.32	tsapp_add_cyclic_msg_can	14
1.4.33	tsapp_add_cyclic_msg_canfd	14
1.4.34	tsapp_delete_cyclic_msg_can	14
1.4.35	tsapp_delete_cyclic_msg_canfd	14
1.4.36	tsapp_delete_cyclic_msgs	14
1.4.37	tsapp_set_vendor_detect_preferences	14
1.4.38	tsapp_get_vendor_detect_preferences	14
1.4.39	tsapp_configure_baudrate_can	15
1.4.40	tsapp_configure_baudrate_canfd	15
1.4.41	tsapp_enable_bus_statistics	15
1.4.42	tsapp_clear_bus_statistics	15
1.4.43	tsapp_get_bus_statistics	15
1.4.44	tsapp_get_fps_can	15
1.4.45	tsapp_get_fps_canfd	15
1.4.46	tsapp_get_fps_lin	16
1.4.47	tsapp_register_event_can	16
1.4.48	tsapp_unregister_event_can	16
1.4.49	tsapp_register_event_canfd	16
1.4.50	tsapp_unregister_event_canfd	16
1.4.51	tsapp_register_event_lin	16
1.4.52	tsapp_unregister_event_lin	16
1.4.53	tsapp_unregister_events_can	16
1.4.54	tsapp_unregister_events_lin	17
1.4.55	tsapp_unregister_events_canfd	17
1.4.56	tsapp_unregister_events_all	17
1.4.57	tsapp_start_logging	17
1.4.58	tsapp_stop_logging	17
1.4.59	tsdb_load_can_db	20
1.4.60	tsdb_unload_can_db	20
1.4.61	tsdb_unload_can_dbs	20
1.4.62	tsdb_get_can_db_count	21
1.4.63	tsdb_get_can_db_id	21
1.4.64	tsdb_get_can_db_info	21

1.4.65	tsdb_set_signal_value_can	21
1.4.66	tsdb_get_signal_value_can	21
1.4.67	tsdb_set_signal_value_canfd	21
1.4.68	tsdb_get_signal_value_canfd	21
1.4.69	tslog_add_online_replay_config	21
1.4.70	tslog_set_online_replay_config	22
1.4.71	tslog_get_online_replay_count	23
1.4.72	tslog_get_online_replay_config	23
1.4.73	tslog_del_online_replay_config	24
1.4.74	tslog_del_online_replay_configs	24
1.4.75	tslog_start_online_replay	24
1.4.76	tslog_start_online_replays	25
1.4.77	tslog_pause_online_replay	25
1.4.78	tslog_pause_online_replays	25
1.4.79	tslog_stop_online_replay	25
1.4.80	tslog_stop_online_replays	25
1.4.81	tslog_get_online_replay_status	26

TSMaster API Introduction

TSMaster DLL library “TSMaster.dll” is the function API interface of all TSMaster functionalities, which can be located in:

{TSMaster installation directory}\bin\Data\SDK\lib\x86\TSMaster.dll

Calling “TSMaster.dll” requires TSMaster program to be installed in your host computer, since many dependencies such as hardware drivers and software drivers of TSMaster are also required by “TSMaster.dll”. After installation of TSMaster application, you can copy “TSMaster.dll” to any location with your host “.exe” application for software development and deployment.

TSMaster API Documentation

1.1 Start Using TSMaster API

Before writing your first application with TSMaster API, you may open and execute existing TSMaster API demo applications to learn the API functionalities. The binary demo applications are located in the following directory, which contains already built applications based on TSMaster.dll:

{TSMaster installation directory}\bin\Data\SDK\examples\bin_x86

You can also view the source code of each demo application in the following directory:

{TSMaster installation directory}\bin\Data\SDK\examples

TSMaster API currently only supports x86 architecture.

1.2 TSMaster API General Information

Each TSMaster API has a return value of integer, indicating error code. Error codes are listed below for error checking:

```
// error code
IDX_ERR_OK                      = 0 ;
IDX_ERR_IDX_OUT_OF_RANGE       = 1 ;
IDX_ERR_CONNECT_FAILED         = 2 ;
IDX_ERR_DEV_NOT_FOUND          = 3 ;
IDX_ERR_CODE_NOT_VALID         = 4 ;
IDX_ERR_ALREADY_CONNECTED      = 5 ;
IDX_ERR_HID_WRITE_FAILED       = 6 ;
IDX_ERR_HID_READ_FAILED        = 7 ;
IDX_ERR_HID_TX_BUFF_OVERRUN    = 8 ;
IDX_ERR_HID_TX_TOO_LARGE       = 9 ;
IDX_ERR_PACKET_ID_INVALID      = 10;
IDX_ERR_PACKET_LEN_INVALID     = 11;
IDX_ERR_INTERNAL_TEST_FAILED   = 12;
IDX_ERR_RX_PACKET_LOST         = 13;
IDX_ERR_HID_SETUP_DI           = 14;
IDX_ERR_HID_CREATE_FILE        = 15;
IDX_ERR_HID_READ_HANDLE        = 16;
IDX_ERR_HID_WRITE_HANDLE       = 17;
IDX_ERR_HID_SET_INPUT_BUFF     = 18;
IDX_ERR_HID_GET_PREPARED       = 19;
IDX_ERR_HID_GET_CAPS           = 20;
IDX_ERR_HID_WRITE_FILE         = 21;
IDX_ERR_HID_GET_OVERLAPPED     = 22;
IDX_ERR_HID_SET_FEATURE        = 23;
IDX_ERR_HID_GET_FEATURE        = 24;
IDX_ERR_HID_DEVICE_IO_CTRL     = 25;
IDX_ERR_HID_SEND_FEATURE_RPT   = 26;
```

```

IDX_ERR_HID_GET_MANU_STR      = 27;
IDX_ERR_HID_GET_PROD_STR     = 28;
IDX_ERR_HID_GET_SERIAL_STR   = 29;
IDX_ERR_HID_GET_INDEXED_STR  = 30;
IDX_ERR_TX_TIMEOUT           = 31;
IDX_ERR_HW_DFU_WRITE_FLASH_FAILED = 32;
IDX_ERR_HW_DFU_WRITE_WO_ERASE = 33;
IDX_ERR_HW_DFU_CRC_CHECK_ERROR = 34;
IDX_ERR_HW_DFU_COMMAND_TIMED_OUT = 35;
IDX_ERR_HW_PACKET_ID_INVALID = 36;
IDX_ERR_HW_PACKET_LEN_INVALID = 37;
IDX_ERR_HW_INTERNAL_TEST_FAILED = 38;
IDX_ERR_HW_RX_FROM_PC_PACKET_LOST = 39;
IDX_ERR_HW_TX_TO_PC_BUFF_OVERRUN = 40;
IDX_ERR_HW_API_PAEAMETER_INVALID = 41;
IDX_ERR_DFU_FILE_LOAD_FAILED = 42;
IDX_ERR_DFU_HEADER_WRITE_FAILED = 43;
IDX_ERR_READ_STATUS_TIMEOUT = 44;
IDX_ERR_CALLBACK_ALREADY_EXISTS = 45;
IDX_ERR_CALLBACK_NOT_EXISTS = 46;
IDX_ERR_FILE_INVALID = 47;
IDX_ERR_DB_ID_NOT_FOUND = 48;
IDX_ERR_SW_API_PAEAMETER_INVALID = 49;
IDX_ERR_SW_API_GENERIC_TIMEOUT = 50;
IDX_ERR_SW_API_SET_CONF_FAILED = 51;
IDX_ERR_SW_API_INDEX_OUT_OF_BOUNDS = 52;
IDX_ERR_SW_API_WAIT_TIMEOUT = 53;
IDX_ERR_SW_API_GET_IO_FAILED = 54;
IDX_ERR_SW_API_SET_IO_FAILED = 55;
IDX_ERR_SW_API_REPLAY_ON_GOING = 56;
IDX_ERR_SW_API_INSTANCE_NOT_EXISTS = 57;
IDX_ERR_HW_CAN_TRANSMIT_FAILED = 58;
IDX_ERR_HW_NO_RESPONSE = 59;
IDX_ERR_SW_CAN_MSG_NOT_FOUND = 60;
IDX_ERR_SW_CAN_RECV_BUFFER_EMPTY = 61;
IDX_ERR_SW_CAN_RECV_PARTIAL_READ = 62;
IDX_ERR_SW_API_LINCONFIG_FAILED = 63;
IDX_ERR_SW_API_FRAMENUM_OUTOFRANGE = 64;
IDX_ERR_SW_API_LDFCONFIG_FAILED = 65;
IDX_ERR_SW_API_LDFCONFIG_CMDERR = 66;
IDX_ERR_SW_ENV_NOT_READY = 67;
IDX_ERR_SECURITY_FAILED = 68;
IDX_ERR_XL_ERROR = 69;
IDX_ERR_SEC_INDEX_OUTOFRANGE = 70;
IDX_SEC_ERR_STRINGLENGTH_OUTOF_RANGE = 71;
IDX_SEC_ERR_KEY_IS_NOT_INITIALIZATION = 72;
IDX_SEC_ERR_KEY_IS_WRONG = 73;
IDX_SEC_ERR_NOT_PERMIT_WRITE = 74;
IDX_SEC_ERR_16BYTES_MULTIPLE = 75;
IDX_ERR_LIN_CHN_OUTOF_RANGE = 76;
IDX_ERR_DLL_NOT_READY = 77;
IDX_ERR_FEATURE_NOT_SUPPORTED = 78;
IDX_ERR_COMMON_SERV_ERROR = 79;
IDX_ERR_READ_PARA_OVERFLOW = 80;
IDX_ERR_INVALID_CHANNEL_MAPPING = 81;
IDX_ERR_TSLIB_GENERIC_OPERATION_FAILED = 82;
IDX_ERR_TSLIB_ITEM_ALREADY_EXISTS = 83;
IDX_ERR_TSLIB_ITEM_NOT_FOUND = 84;
IDX_ERR_TSLIB_LOGICAL_CHANNEL_INVALID = 85;
IDX_ERR_FILE_NOT_EXISTS = 86;
IDX_ERR_NO_INIT_ACCESS = 87;
IDX_ERR_CHN_NOT_ACTIVE = 88;
IDX_ERR_CHN_NOT_CREATED = 89;
IDX_ERR_APPNAME_LENGTH_OUT_OF_RANGE = 90;
IDX_ERR_PROJECT_IS_MODIFIED = 91;
IDX_ERR_SIGNAL_NOT_FOUND_IN_DB = 92;
IDX_ERR_MESSAGE_NOT_FOUND_IN_DB = 93;
IDX_ERR_TSMaster_IS_NOT_INSTALLED = 94;
IDX_ERR_LIB_LOAD_FAILED = 95;

```

```

IDX_ERR_LIB_FUNCTION_NOT_FOUND      = 96;
IDX_ERR_LIB_NOT_INITIALIZED          = 97;
IDX_ERR_PCAN_GENRIC_ERROR            = 98;
IDX_ERR_KVASER_GENERIC_ERROR         = 99;
IDX_ERR_ZLG_GENERIC_ERROR            = 100;
IDX_ERR_ICS_GENERIC_ERROR            = 101;
IDX_ERR_TC1005_GENERIC_ERROR         = 102;
IDX_ERR_SYSTEM_VAR_NOT_FOUND         = 103;
IDX_ERR_INCORRECT_SYSTEM_VAR_TYPE    = 104;
IDX_ERR_CYCLIC_MSG_NOT_EXIST         = 105;
IDX_ERR_BAUD_NOT_AVAIL               = 106;
IDX_ERR_DEV_NOT_SUPPORT_SYNC_SEND    = 107;
IDX_ERR_MP_WAIT_TIME_NOT_SATISFIED   = 108;
IDX_ERR_CANNOT_OPERATE_WHILE_CONNECTED = 109;
IDX_ERR_CREATE_FILE_FAILED           = 110;
IDX_ERR_PYTHON_EXECUTE_FAILED        = 111;
IDX_ERR_SIGNAL_MULTIPLEXED_NOT_ACTIVE = 112;
IDX_ERR_GET_HANDLE_BY_CHANNEL_FAILED  = 113;

```

Error code descriptions can be found in the following list:

0	IDX_ERR_OK	OK
1	IDX_ERR_IDX_OUT_OF_RANGE	Index out of range
2	IDX_ERR_CONNECT_FAILED	Connect failed
3	IDX_ERR_DEV_NOT_FOUND	Device not found
4	IDX_ERR_CODE_NOT_VALID	Error code not valid
5	IDX_ERR_ALREADY_CONNECTED	HID device already connected
6	IDX_ERR_HID_WRITE_FAILED	HID write data failed
7	IDX_ERR_HID_READ_FAILED	HID read data failed
8	IDX_ERR_HID_TX_BUFF_OVERRUN	HID TX buffer overrun
9	IDX_ERR_HID_TX_TOO_LARGE	HID TX buffer too large
10	IDX_ERR_PACKET_ID_INVALID	HID RX packet report ID invalid
11	IDX_ERR_PACKET_LEN_INVALID	HID RX packet length invalid
12	IDX_ERR_INTERNAL_TEST_FAILED	Internal test failed
13	IDX_ERR_RX_PACKET_LOST	RX packet lost
14	IDX_ERR_HID_SETUP_DI	SetupDiGetDeviceInterfaceDetail
15	IDX_ERR_HID_CREATE_FILE	Create file failed
16	IDX_ERR_HID_READ_HANDLE	CreateFile failed for read handle
17	IDX_ERR_HID_WRITE_HANDLE	CreateFile failed for write handle
18	IDX_ERR_HID_SET_INPUT_BUFF	HidD_SetNumInputBuffers
19	IDX_ERR_HID_GET_PREPARED	HidD_GetPreparedData
20	IDX_ERR_HID_GET_CAPS	HidP_GetCaps
21	IDX_ERR_HID_WRITE_FILE	WriteFile
22	IDX_ERR_HID_GET_OVERLAPPED	GetOverlappedResult
23	IDX_ERR_HID_SET_FEATURE	HidD_SetFeature
24	IDX_ERR_HID_GET_FEATURE	HidD_GetFeature
25	IDX_ERR_HID_DEVICE_IO_CTRL	Send Feature Report DeviceIoControl
26	IDX_ERR_HID_SEND_FEATURE_RPT	Send Feature Report GetOverLappedResult
27	IDX_ERR_HID_GET_MANU_STR	HidD_GetManufacturerString
28	IDX_ERR_HID_GET_PROD_STR	HidD_GetProductString
29	IDX_ERR_HID_GET_SERIAL_STR	HidD_GetSerialNumberString

30	IDX_ERR_HID_GET_INDEXED_STR	HidD_GetIndexedString
31	IDX_ERR_TX_TIMEOUT	Transmit timed out
32	IDX_ERR_HW_DFU_WRITE_FLASH_FAILED	HW DFU flash write failed
33	IDX_ERR_HW_DFU_WRITE_WO_ERASE	HW DFU write without erase
34	IDX_ERR_HW_DFU_CRC_CHECK_ERROR	HW DFU crc check error
35	IDX_ERR_HW_DFU_RESET_BEFORE_CRC	HW DFU reset before crc check success
36	IDX_ERR_HW_PACKET_ID_INVALID	HW packet identifier invalid
37	IDX_ERR_HW_PACKET_LEN_INVALID	HW packet length invalid
38	IDX_ERR_HW_INTERNAL_TEST_FAILED	HW internal test failed
39	IDX_ERR_HW_RX_FROM_PC_PACKET_LOST	HW rx from pc packet lost
40	IDX_ERR_HW_TX_TO_PC_BUFF_OVERRUN	HW tx to pc buffer overrun
41	IDX_ERR_HW_API_PAEAMETER_INVALID	HW API parameter invalid
42	IDX_ERR_DFU_FILE_LOAD_FAILED	DFU file load failed
43	IDX_ERR_DFU_HEADER_WRITE_FAILED	DFU header write failed
44	IDX_ERR_READ_STATUS_TIMEOUT	Read status timed out
45	IDX_ERR_CALLBACK_ALREADY_EXISTS	Callback already exists
46	IDX_ERR_CALLBACK_NOT_EXISTS	Callback not exists
47	IDX_ERR_FILE_INVALID	File corrupted or not recognized
48	IDX_ERR_DB_ID_NOT_FOUND	Database unique id not found
49	IDX_ERR_SW_API_PAEAMETER_INVALID	Software API parameter invalid
50	IDX_ERR_SW_API_GENERIC_TIMEOUT	Software API generic timed out
51	IDX_ERR_SW_API_SET_CONF_FAILED	Software API set hw config. failed
52	IDX_ERR_SW_API_INDEX_OUT_OF_BOUNDS	Index out of bounds
53	IDX_ERR_SW_API_WAIT_TIMEOUT	RX wait timed out
54	IDX_ERR_SW_API_GET_IO_FAILED	Get I/O failed
55	IDX_ERR_SW_API_SET_IO_FAILED	Set I/O failed
56	IDX_ERR_SW_API_REPLAY_ON_GOING	An active replay is already running
57	IDX_ERR_SW_API_INSTANCE_NOT_EXISTS	Instance not exists
58	IDX_ERR_HW_CAN_TRANSMIT_FAILED	CAN message transmit failed
59	IDX_ERR_HW_NO_RESPONSE	No response from hardware
60	IDX_ERR_SW_CAN_MSG_NOT_FOUND	CAN message not found
61	IDX_ERR_SW_CAN_RECV_BUFFER_EMPTY	User CAN receive buffer empty
62	IDX_ERR_SW_CAN_RECV_PARTIAL_READ	CAN total receive count <> desired count
63	IDX_ERR_SW_API_LINCONFIG_FAILED	LIN config failed
64	IDX_ERR_SW_API_FRAMENUM_OUTOFRANGE	LIN frame number out of range
65	IDX_ERR_SW_API_LDFCONFIG_FAILED	LDF config failed
66	IDX_ERR_SW_API_LDFCONFIG_CMDERR	LDF config cmd error
67	IDX_ERR_SW_ENV_NOT_READY	TSMaster envrionment not ready
68	IDX_ERR_RESERVED_FAILED	reserved failed
69	IDX_ERR_XL_ERROR	XL driver error
70	IDX_ERR_SEC_INDEX_OUTOFRANGE	index out of range
71	IDX_SEC_ERR_STRINGLENGTH_OUTFOF_RANGE	string length out of range

72	IDX_SEC_ERR_KEY_IS_NOT_INITIALIZATION	key is not initialized
73	IDX_SEC_ERR_KEY_IS_WRONG	key is wrong
74	IDX_SEC_ERR_NOT_PERMIT_WRITE	write not permitted
75	IDX_SEC_ERR_16BYTES_MULTIPLE	16 bytes multiple
76	IDX_ERR_LIN_CHN_OUTOF_RANGE	LIN channel out of range
77	IDX_ERR_DLL_NOT_READY	DLL not ready
78	IDX_ERR_FATURE_NOT_SUPPORTED	Feature not supported
79	IDX_ERR_COMMON_SERV_ERROR	common service error
80	IDX_ERR_READ_PARA_OVERFLOW	read parameter overflow
81	IDX_ERR_INVALID_CHANNEL_MAPPING	Invalid application channel mapping
82	IDX_ERR_TSLIB_GENERIC_OPERATION_FAILED	libTSMaster generic operation failed
83	IDX_ERR_TSLIB_ITEM_ALREADY_EXISTS	item already exists
84	IDX_ERR_TSLIB_ITEM_NOT_FOUND	item not found
85	IDX_ERR_TSLIB_LOGICAL_CHANNEL_INVALID	logical channel invalid
86	IDX_ERR_FILE_NOT_EXISTS	file not exists
87	IDX_ERR_NO_INIT_ACCESS	no init access cannot set baudrate
88	IDX_ERR_CHN_NOT_ACTIVE	the channel is inactive
89	IDX_ERR_CHN_Not_Created	the channel is not created
90	IDX_ERR_APPNAME_LENGTH_OUT_OF_RANGE	length of the appname is out of range
91	IDX_ERR_PROJECT_IS_MODIFIED	project is modified
92	IDX_ERR_SIGNAL_NOT_FOUND_IN_DB	signal not found in database
93	IDX_ERR_MESSAGE_NOT_FOUND_IN_DB	message not found in database
94	IDX_ERR_TSMaster_IS_NOT_INSTALLED	TSMaster is not installed
95	IDX_ERR_LIB_LOAD_FAILED	Library load failed
96	IDX_ERR_LIB_FUNCTION_NOT_FOUND	Library function not found
97	IDX_ERR_LIB_NOT_INITIALIZED	cannot find libTSMaster.dll, use "set_libtmaster_location" to set its location before calling initialize_lib_tsmaster
98	IDX_ERR_PCAN_GENERIC_ERROR	PCAN generic operation error
99	IDX_ERR_KVASER_GENERIC_ERROR	Kvaser generic operation error
100	IDX_ERR_ZLG_GENERIC_ERROR	ZLG generic operation error
101	IDX_ERR_ICS_GENERIC_ERROR	ICS generic operation error
102	IDX_ERR_TC1005_GENERIC_ERROR	TC1005 generic operation error
103	IDX_ERR_SYSTEM_VAR_NOT_FOUND	System variable not found
104	IDX_ERR_INCORRECT_SYSTEM_VAR_TYPE	Incorrect system variable type
105	IDX_ERR_CYCLIC_MSG_NOT_EXIST	Message not existing, update failed
106	IDX_ERR_BAUD_NOT_AVAIL	Specified baudrate not available
107	IDX_ERR_DEV_NOT_SUPPORT_SYNC_SEND	Device does not support sync. transmit
108	IDX_ERR_MP_WAIT_TIME_NOT_SATISFIED	Wait time not satisfied
109	IDX_ERR_CANNOT_OPERATE_WHILE_CONNECTED	Cannot operate while app is connected
110	IDX_ERR_CREATE_FILE_FAILED	Create file failed
111	IDX_ERR_PYTHON_EXECUTE_FAILED	Execute python failed

112	IDX_ERR_SIGNAL_MULTIPLEXED_NOT_ACTIVE	Current multiplexed signal is not active
113	IDX_ERR_GET_HANDLE_BY_CHANNEL_FAILED	Get handle by logic channel failed

1.3 TSMaster API basic data types

```

typedef enum {lvlError = 1, lvlWarning = 2, lvlOK = 3, lvlHint = 4, lvlInfo = 5, lvlVerbose = 6} TLogLevel;
// basic var type definition
typedef unsigned __int8 u8;
typedef signed __int8 s8;
typedef unsigned __int16 u16;
typedef signed __int16 s16;
typedef unsigned __int32 u32;
typedef signed __int32 s32;
typedef unsigned __int64 u64;
typedef signed __int64 s64;
// pointer definition
typedef unsigned __int8* pu8;
typedef signed __int8* ps8;
typedef unsigned __int16* pu16;
typedef signed __int16* ps16;
typedef unsigned __int32* pu32;
typedef signed __int32* ps32;
typedef unsigned __int64* pu64;
typedef signed __int64* ps64;
typedef float* pfloat;
typedef double* pdouble;

#define APP_DEVICE_NAME_LENGTH 32
typedef enum {BUS_UNKNOWN_TYPE = 0, TS_TCP_DEVICE = 1, XL_USB_DEVICE = 2, TS_USB_DEVICE = 3} TLIBBusToolDeviceType;
typedef enum {APP_CAN = 0, APP_LIN = 1} TLIBApplicationChannelType;
typedef enum {
    cbsBusLoad = 0, cbsPeakLoad, cbsFpsStdData, cbsAllStdData,
    cbsFpsExtData, cbsAllExtData, cbsFpsStdRemote, cbsAllStdRemote,
    cbsFpsExtRemote, cbsAllExtRemote, cbsFpsErrorFrame, cbsAllErrorFrame
} TLIBCANBusStatistics;
typedef enum {fdtCAN = 0, fdtISOCANFD = 1, fdtNonISOCANFD = 2} TCANFDControllerType;
typedef enum {fdmNormal = 0, fdmACKOff = 1, fdmRestricted = 2} TCANFDControllerMode;

typedef enum {ortImmediately = 0, ortAsLog = 1, ortDelayed = 2} TLIBOnlineReplayTimingMode;
typedef enum {orsNotStarted = 0, orsRunning = 1, orsPaused = 2, orsCompleted = 3, orsTerminated = 4} TLIBOnlineReplayStatus;

```

1.4 TSMaster API Description

1.4.1 initialize_lib_tsmaster

1.4.2 finalize_lib_tsmaster

1.4.3 tsapp_set_current_application

1.4.4 tsapp_get_current_application

1.4.5 tsapp_del_application

1.4.6 tsapp_add_application

1.4.7 tsapp_set_logger

1.4.8 tsapp_get_application_list

1.4.9 tsapp_set_can_channel_count

1.4.10 tsapp_set_lin_channel_count

1.4.11 tsapp_get_can_channel_count

1.4.12 tsapp_get_lin_channel_count

1.4.13 tsapp_set_mapping

1.4.14 tsapp_get_mapping

1.4.15 tsapp_del_mapping

1.4.16 tsapp_connect

1.4.17 tsapp_disconnect

1.4.18 tsapp_set_turbo_mode

1.4.19 tsapp_get_turbo_mode

1.4.20 tsapp_get_error_description

1.4.21 tsapp_show_channel_mapping_window

1.4.22 tsapp_show_hardware_configuration_window

1.4.23 tsapp_show_tsmaster_window

1.4.24 tsapp_get_timestamp

1.4.25 tsapp_transmit_can_async

1.4.26 tsapp_transmit_canfd_async

1.4.27 tsapp_transmit_lin_async

1.4.28 tsapp_transmit_fastlin_async

1.4.29 tsapp_transmit_can_sync

1.4.30 tsapp_transmit_canfd_sync

1.4.31 tsapp_transmit_lin_sync

1.4.32 tsapp_add_cyclic_msg_can

1.4.33 tsapp_add_cyclic_msg_canfd

1.4.34 tsapp_delete_cyclic_msg_can

1.4.35 tsapp_delete_cyclic_msg_canfd

1.4.36 tsapp_delete_cyclic_msgs

1.4.37 tsapp_set_vendor_detect_preferences

1.4.38 tsapp_get_vendor_detect_preferences

1.4.39 tsapp_configure_baudrate_can

1.4.40 tsapp_configure_baudrate_canfd

1.4.41 tsapp_enable_bus_statistics

1.4.42 tsapp_clear_bus_statistics

1.4.43 tsapp_get_bus_statistics

1.4.44 tsapp_get_fps_can

1.4.45 tsapp_get_fps_canfd

1.4.46 tsapp_get_fps_lin

1.4.47 tsapp_register_event_can

1.4.48 tsapp_unregister_event_can

1.4.49 tsapp_register_event_canfd

1.4.50 tsapp_unregister_event_canfd

1.4.51 tsapp_register_event_lin

1.4.52 tsapp_unregister_event_lin

1.4.53 tsapp_unregister_events_can

1.4.54 tsapp_unregister_events_lin

1.4.55 tsapp_unregister_events_canfd

1.4.56 tsapp_unregister_events_all

1.4.57 tsapp_start_logging

1.4.58 tsapp_stop_logging

1.4.59 ini_create

```
s32 h;  
if (0 == app.ini_create("c:\\lc_ramdisk\\a.ini", &h)){  
    log("ini object created: %x", h);  
}  
  
app.ini_write_int32(h, "sec1", "val32", 32);  
app.ini_write_int64(h, "sec1", "val64", 64);
```

```
app.ini_write_bool(h, "sec1", "valb", true);
app.ini_write_float(h, "sec1", "valf", 1.234);
app.ini_write_string(h, "sec1", "vals", "1.234");

s32 i32;
s64 i64;
bool b;
double d;
char s[512];
s32 l = 512;

if (0 == app.ini_read_int32(h, "sec1", "val32", &i32, 32)){
    log("i32 = %d", i32);
}
if (0 == app.ini_read_int64(h, "sec1", "val64", &i64, 64)){
    log("i64 = %lld", i64);
}
if (0 == app.ini_read_bool(h, "sec1", "valb", &b, true)){
    if (b){
        log("true");
    } else {
        log("false");
    }
}
if (0 == app.ini_read_float(h, "sec1", "valf", &d, 1.234)){
    log("f = %f", d);
}
if (0 == app.ini_read_string(h, "sec1", "vals", s, &l, "1.234")){
    log("str = %s", s);
}
if (0 == app.ini_section_exists(h, "sec1")){
    log("sec1 exists");
} else {
    log("sec1 not exists");
}
app.ini_delete_key(h, "sec1", "val32");
app.ini_delete_section(h, "sec1");

if (0 == app.ini_close(h)) {
    log("ini object closed: %x", h);
}
```

1.4.60 ini_write_int32

1.4.61 ini_write_int64

1.4.62 ini_write_bool

1.4.63 ini_write_float

1.4.64 ini_write_string

1.4.65 ini_read_int32

1.4.66 ini_read_int64

1.4.67 ini_read_bool

1.4.68 ini_read_float

1.4.69 ini_read_string

1.4.70 ini_section_exists

1.4.71 ini_key_exists

1.4.72 ini_delete_key

1.4.73 ini_delete_section

1.4.74 ini_close

1.4.75 tsdb_load_can_db

1.4.76 tsdb_unload_can_db

1.4.77 tsdb_unload_can_dbs

1.4.78 tsdb_get_can_db_count

1.4.79 tsdb_get_can_db_id

1.4.80 tsdb_get_can_db_info

1.4.81 tsdb_set_signal_value_can

1.4.82 tsdb_get_signal_value_can

1.4.83 tsdb_set_signal_value_canfd

1.4.84 tsdb_get_signal_value_canfd

1.4.85 tslog_add_online_replay_config

To create an online replay engine from log file.

<pre>typedef s32 (__stdcall* tslog_add_online_replay_config)(const char* AFileName, s32* AIndex);</pre>	
AFileName	Blf File Name, can be absolute path or relative path, when using relative path, the path is relative to "TSMaster.exe" in your current installation directory. For example, the TSMaster.exe is located in the following directory: C:\Program Files (x86)\TOSUN\TSMaster\bin\ And if you want to use a blf file in "C:\Program Files (x86)\TOSUN\TSMaster\bin\Data\Demo\Logs\CANSystem_CAN2.blf", you can pass relative path as: ".\Data\Demo\Logs\CANSystem_CAN2.blf"
AIndex	Internal online replay engine index will be returned if this online replay configuration is successfully created

1.4.86 tslog_set_online_replay_config

Set the properties of online replay engine by engine index.

<pre>typedef s32 (__stdcall* tslog_set_online_replay_config)(const s32 AIndex, const char* AName, const char* AFileName, const bool AAutoStart, const bool AIsRepetitiveMode, const TLIBOnlineReplayTimingMode AStartTimingMode, const s32 AStartDelayTimeMs, const bool ASendTx, const bool ASendRx, const char* AMappings);</pre>	
AIndex	Replay engine index
AName	Replay engine name
AFileName	File name of the blf to be replayed
AAutoStart	Whether the log file is automatically replayed after application is connected
AIsRepetitiveMode	True: the log file will be replayed continuously False: the log file will be replayed only once
AStartTimingMode	ortImmediately(0): The first frame in the log file will be directly sent when replay is started ortAsLog(1): The first frame is sent according to log file ortDelayed(2): The first frame will be sent after a certain delay specified in the parameter
AStartDelayTimeMs	The first frame will be sent after the specified delay time in milliseconds

ASendTx	Frames marked as "Tx" in the log file will be replayed
ASendRx	Frames marked as "Rx" in the log file will be replayed
AMappings	Mappings of each channel is specified in a string separated by ",", and channel 0 means ignore, for example, direct mapping of each channel is: "1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32" If Channel 1 is mapped to Channel 3, the parameter is: "3,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32"

1.4.87 tslog_get_online_replay_count

Get online replay engine count.

<pre>typedef s32 (__stdcall* tslog_get_online_replay_count)(s32* ACount);</pre>	
ACount	Return the online replay engine count

1.4.88 tslog_get_online_replay_config

Get online replay engine properties by engine index.

<pre>typedef s32 (__stdcall* tslog_get_online_replay_config)(const s32 AIndex, char** AName, char** AFileName, bool* AAutoStart, bool* AIsRepetitiveMode, TLIBOnlineReplayTimingMode* AStartTimingMode, s32* AStartDelayTimeMs, bool* ASendTx, bool* ASendRx, char** AMappings);</pre>	
AIndex	Replay engine index
AName	Replay engine name
AFileName	File name of the blf to be replayed
AAutoStart	Whether the log file is automatically replayed after application is connected
AIsRepetitiveMode	True: the log file will be replayed continuously False: the log file will be replayed only once
AStartTimingMode	ortImmediately(0): The first frame in the log file will be directly sent when replay is started ortAsLog(1): The first frame is sent according to log file

	ortDelayed(2): The first frame will be sent after a certain delay specified in the parameter
AStartDelayTimeMs	The first frame will be sent after the specified delay time in milliseconds
ASendTx	Frames marked as "Tx" in the log file will be replayed
ASendRx	Frames marked as "Rx" in the log file will be replayed
AMappings	Mappings of each channel is specified in a string separated by ",", and channel 0 means ignore, for example, direct mapping of each channel is: "1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32" If Channel 1 is mapped to Channel 3, the parameter is: "3,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32"

1.4.89 tslog_del_online_replay_config

Delete an online replay engine by its engine index.

typedef s32 (__stdcall* tslog_del_online_replay_config) (const s32 AIndex);	
AIndex	Replay engine index

1.4.90 tslog_del_online_replay_configs

Delete all online replay engines.

typedef s32 (__stdcall* tslog_del_online_replay_configs) (void);	

1.4.91 tslog_start_online_replay

Start online replay by engine index.

Note: The application should be connected before engine start.

typedef s32 (__stdcall* tslog_start_online_replay) (const s32 AIndex);	
AIndex	Replay engine index

1.4.92 tslog_start_online_replays

Start all online replay engines.

```
typedef s32 (__stdcall* tslog_start_online_replays) (void);
```

1.4.93 tslog_pause_online_replay

Pause an online replay engine by its engine index.

```
typedef s32 (__stdcall* tslog_pause_online_replay) (const s32 AIndex);
```

AIndex	Replay engine index
--------	---------------------

1.4.94 tslog_pause_online_replays

Pause all online replay engines.

```
typedef s32 (__stdcall* tslog_pause_online_replays) (void);
```

1.4.95 tslog_stop_online_replay

Stop an online replay engine by its engine index.

```
typedef s32 (__stdcall* tslog_stop_online_replay) (const s32 AIndex);
```

AIndex	Replay engine index
--------	---------------------

1.4.96 tslog_stop_online_replays

Stop all online replay engines.

```
typedef s32 (__stdcall* tslog_stop_online_replays) (void);
```

1.4.97 tslog_get_online_replay_status

Get the replay status of specific online replay engine by its engine index.

```
typedef s32 ( __stdcall* tslog_get_online_replay_status)(const s32 AIndex, TLIBOnline  
ReplayStatus* AStatus, float* AProgressPercent100);
```

AIndex	Replay engine index
AStatus	orsNotStarted = 0: The replay engine has not been started orsRunning = 1: The replay engine is running orsPaused = 2: The replay engine has been paused orsCompleted = 3: The replay engine has replayed all messages and entered stop mode orsTerminated = 4: The replay engine encounters errors and entered stop mode
AProgressPercent100	Process in percentage, from 0 to 100 %