

Peripheral Service API specification

**Note/
CWORDXX is mask word for confidential information.**

2019.10.17
TOYOTA MOTOR CORPRATION

Module Documentation

BaseSystem

[Peripheral service](#)

Detailed Description

Peripheral_service

[Communication](#)

Macros

```
#define SERVICE\_PSMShadow "PS_PSMShadow"  
#define SERVICE\_IPC "PS_IPC_DISP"  
#define SERVICE\_IPC\_DISP "PS_IPC"  
#define SERVICE\_IPCTESTAPP "PS_IPC_TestApp"  
#define SERVICE\_MP\_SHADOW "PS_IPC_MP_Shadow"  
#define SERVICE\_SENSORShadow "PS_SensorShadow"  
#define SERVICE\_CWORD33\_DEBUGDUMPTEST "PS_CWORD33_DebugDumpTest"  
#define SERVICE\_PSMShadowTEST "PS_TEST_PSMShadow"  
#define SERVICE\_LOGGERShadow "PS_LoggerShadow"  
#define SERVICE\_SOFTWAREUPDATESHADOW "PS_SoftwareUpdateShadow"
```

Detailed Description

Macro Definition Documentation

#define SERVICE_CWORD33_DEBUGDUMPTEST "PS_CWORD33_DebugDumpTest"

Define the name of SERVICE_CWORD33_DEBUGDUMPTEST

#define SERVICE_IPC "PS_IPC_DISP"

Define the name of SERVICE_IPC

#define SERVICE_IPC_DISP "PS_IPC"

Define the name of SERVICE_IPC_DISP

#define SERVICE_IPCTESTAPP "PS_IPC_TestApp"

Define the name of SERVICE_IPCTESTAPP

#define SERVICE_LOGGERSHADOW "PS_LoggerShadow"

Define the name of SERVICE_LOGGERSHADOW

#define SERVICE_MP_SHADOW "PS_IPC_MP_Shadow"

Define the name of SERVICE_MP_SHADOW

#define SERVICE_PSMshadow "PS_PSMShadow"

Define the name of SERVICE_PSMshadow

#define SERVICE_PSMshadowTEST "PS_TEST_PSMShadow"

Define the name of SERVICE_PSMshadowTEST

#define SERVICE_SENSORshadow "PS_SensorShadow"

Define the name of SERVICE_SENSORshadow

#define SERVICE_SOFTWAREUPDATESHADOW "PS_SoftwareUpdateShadow"

Define the name of SERVICE_SOFTWAREUPDATESHADOW

Communication

[CAN](#)

[LAN](#)

Detailed Description

CAN

struct [CAN MSG CANDATA DAT](#)
struct [CAN MSG CANDATA](#)
struct [CAN MSG CANGWDATA DAT](#)
struct [CAN MSG CAN_CWORD29 DATA](#)
struct [CAN MSG CANGWDATA](#)
struct [CAN DATA](#)
struct [CAN MSG_CWORD37 DATA DAT](#)
struct [CAN MSG_CWORD37 DATA](#)
struct [_CWORD37_DATA](#)
struct [CAN TRANSMISSION START MSG DAT](#)
struct [CAN_CWORD30_TRANS START MSG DAT](#)
struct [CAN MSG SENDSTS DAT](#)
struct [CAN MSG SENDSTS](#)
struct [CAN MSG_CWORD29_SENDSSTS DAT](#)
struct [CAN MSG_CWORD29_SENDSSTS](#)
struct [CAN_FREQ_TRANS_STOP MSG DAT](#)
struct [CAN MSG COMM_WATCHSTS DAT](#)
struct [CAN MSG COMM_WATCHSTS](#)
struct [CAN DELIVERY ENTRY](#)
struct [CAN_CWORD29_DELIVERY ENTRY](#)
struct [CAN COMM WATCH MSG DAT](#)
struct [CAN_CWORD30_ERRMSG DAT](#)
struct [CAN_CWORD30_ERRMSG](#)
struct [CAN CMD CTRL MSG DAT](#)
struct [CAN MSG CANCMD DAT](#)
struct [CAN MSG CANCMD](#)
struct [CAN CMD MSG SENDSTS DAT](#)
struct [CAN CMD MSG SENDSTS](#)
struct [CAN MSG DATA](#)
struct [_CWORD37_TRANSMISSION START MSG DAT](#)
struct [_CWORD37_MSG DATA](#)
struct [_CWORD37_ERRMSG DAT](#)
struct [_CWORD37_ERRMSG](#)
struct [CAN COMM WATCHEXT MSG DAT](#)
struct [CAN DATA MASK](#)
struct [CAN DATA BIT](#)
struct [CAN_TRANS START MSG DAT](#)
struct [CAN MSG_CWORD34 DATA](#)
struct [_CWORD34_MSG SENDSTS](#)
struct [_CWORD34_MSG COMM WATCHSTS](#)

Macros

```
#define CAN_DEBUG
#define LAN_SERVICE_CAN "CAN_COM_PROT"
#define LAN_SERVICE_CWORD34 "_CWORD34_COM_PROT"
#define LAN_SERVICE_CWORD37 "_CWORD37_COM_PROT"
#define NTFY_Communication_CAN_ISAVAILABLE "Communication/CanIsAvailable"
#define
    NTFY_Communication_CWORD34_ISAVAILABLE "Communication/_CWORD34_IsAvailable"
#define
    NTFY_Communication_CWORD37_ISAVAILABLE "Communication/_CWORD37_IsAvailable"
#define CANIF_RET_NORMAL 1
#define CANIF_RET_ERROR_PARAM 2
#define CANIF_RET_ERROR_BUFFULL 3
#define CANIF_RET_ERROR_CANIDFULL 4
#define CANIF_RET_ERROR_UNDEF 5
#define CANIF_RET_ERROR_PID 6
#define CANIF_RET_ERROR_TIMER 7
#define CANIF_RET_ERROR_CREATE_EVENT 8
#define CANIF_RET_ERROR_CANCEL 9
#define CANIF_RET_ERROR_NULL 10
#define CAN_CMDID_FUELCALC_RST_REQ_DELIVERY 0x00
#define CAN_CMDID_STARTUP_FIN_REQ_TX 0x01
#define CAN_CMDID_MRST_INFO_REQ_TX 0x02
#define CAN_CMDID_VERSION_REQ_TX 0x03
#define CAN_CMDID_CONNECTION_NODE_REQ_TX 0x04
#define CAN_CMDID_BUS_STATUS_REQ_TX 0x05
#define CAN_CMDID_FUELCALC_REQ_TX 0x06
#define CAN_CMDID_FUELCALC_RST_REQ_RX 0x07
#define CAN_CMDID_STARTUP_FIN_RESP_RX 0x08
#define CAN_CMDID_MRST_INFO_RESP_RX 0x09
#define CAN_CMDID_VERSION_RESP_RX 0x0A
#define CAN_CMDID_CONNECTION_NODE_RESP_RX 0x0B
#define CAN_CMDID_BUS_STATUS_RESP_RX 0x0C
#define CWORD37_CMDID_BUS_STATUS_REQ_TX 0x20
#define CWORD37_CMDID_BUS_STATUS_RESP_RX 0x21
#define CAN_DELIVERY_STOP 0
#define CAN_DELIVERY_RESTART 1
#define CAN_SUCCESS 0
#define CAN_RETRYOUT 1
#define CAN_BUFFERFUL 2
#define CAN_CWORD30_ERR_NONREQ 0
#define CAN_CWORD30_RXERR_REQ 1
#define CAN_CWORD30_TXERR_REQ 2
#define CWORD37_ERR_NONREQ 0
#define CWORD37_TXERR_REQ 1
#define CAN_IG_COOPERATION_OFF 0
#define CAN_IG_COOPERATION_ON 1
#define CAN_NTA_INVALID 0xFF
```

```

#define CAN\_CWORD30\_N TA NONE (0x00)
#define CWORD37\_LOGICID MASK 0x3FF
#define CWORD37\_SENDID MASK 0x0F
#define CWORD37\_RCVID MASK 0x0F
#define CWORD37\_CMDID MASK 0x7FF
#define CWORD37\_LOGICID SHIFT 19
#define CWORD37\_SENDID SHIFT 15
#define CWORD37\_RCVID SHIFT 11
#define CWORD37\_CMDID SHIFT 0
#define CWORD37\_ERROR CANID 0xFFFFFFFF
#define CAN RID NOTUSE CODE 0xFF
#define CAN\_CWORD30\_SHARED MEM NAME ("CAN_CWORD30_SHARE_MEMORY")
#define CAN\_CWORD30\_SHARED MEM SZ (1024)
#define CAN\_CWORD30\_SEMAPHO NAME ("CAN_CWORD30_SEMAPHO")
#define CAN\_CWORD30\_SHM NAME "/CAN_CWORD30_SHM_%d"
#define CWORD37\_TRANSMISSION\_SHM NAME "/_CWORD37_TRANSMISSION_SHM_%d"
#define CAN DATA SIZE 63
#define CAN TXDATA SIZE 8
#define CAN\_CWORD30\_TXDATA SIZE CAN\_CWORD30\_SHARED MEM SZ
#define CWORD37\_CANID AREA SIZE 4
#define CWORD37\_DLC AREA SIZE 2
#define CWORD37\_DATAMAX SIZE 2048
#define CAN DELIVERY CANID ENTRY MAX 125
#define CAN DELIVERY OPC ENTRY MAX 255
#define CAN MSGBUF MAX SIZE 528
#define CAN MSGBUF HEADER SIZE 16
#define CAN MSGBUF DATAMAX SIZE \(CAN MSGBUF MAX SIZE - CAN MSGBUF HEADER SIZE\)
#define CWORD37\_MSGBUF MAX SIZE 2076
#define CWORD37\_MSGBUF DATAMAX SIZE \(CWORD37\_MSGBUF MAX SIZE -
    CAN MSGBUF HEADER SIZE\)
#define CANCMD DAT MAX 68
#define CWORD34\_SND\_NUM MAX 255
#define CWORD34\_DLC MAX SIZE 0x08
#define CANGW\_SND\_NUM MAX 0x16
#define CANGW\_DLC MAX SIZE 0x08
#define CAN\_CWORD29\_MEM\_SZ (4100)

```

Typedefs

```

typedef enum \_PS\_CommunicationProtocol PS\_CommunicationProtocol
typedef enum \_PS\_Communication\_InternalProtocol PS\_Communication\_InternalProtocol
typedef enum \_PS\_Communication\_Internal\_DebugProtocol
    PS\_Communication\_Internal\_DebugProtocol
typedef int32_t CANIF\_RET\_API
typedef uint32_t CANID
typedef uint32_t DID

```

Enumerations

```
enum \_PS\_CommunicationProtocol { CID\_CAN\_DATA\_DELIVERY = ***, CID\_CAN\_COMM\_STOP,  
CID\_CAN\_COMM\_RESTORE, CID\_CAN\_TX\_RESULT, CID\_CAN\_CWORD29\_TX\_RESULT,  
CID\_CAN\_CWORD30\_DELIVERY, CID\_CAN\_DIAG\_CWORD29\_DELIVERY,  
CID\_CAN\_CWORD30\_FF\_DELIVERY, CID\_CAN\_CWORD30\_RXERR,  
CID\_CAN\_CWORD30\_TXERR, CID\_CAN\_CMD\_DELIVERY, CID\_CAN\_CMD\_TX\_RESULT,  
CID\_CAN\_CANGW\_DELIVERY, CID\_CWORD37\_DATA\_DELIVERY = ***,  
CID\_CWORD37\_CMD\_DELIVERY, CID\_CWORD37\_COMM\_STOP,  
CID\_CWORD37\_COMM\_RESTORE, CID\_CWORD37\_TX\_ERROR,  
CID\_CWORD34\_DATA\_DELIVERY = ***, CID\_CWORD34\_COMM\_STOP,  
CID\_CWORD34\_COMM\_RESTORE, CID\_CWORD34\_SEND\_RES }  
enum \_PS\_Communication\_InternalProtocol { CID\_CANIF\_DELIVERY\_ENTRY = ***,  
CID\_CANIF\_CWORD29\_DELIVERY\_ENTRY, CID\_CANIF\_TX\_START, CID\_CANIF\_TX\_STOP,  
CID\_CANIF\_COMM\_WATCH, CID\_CANIF\_CWORD30\_TX\_START,  
CID\_CANIF\_CWORD30\_TX\_SEQ\_FIN, CID\_CANIF\_CWORD29\_TX\_START,  
CID\_CANIF\_CMD\_CTRL, CID\_CANIF\_TX\_BIT\_START, CID\_CANIF\_TX\_BIT\_STOP,  
CID\_CWORD37\_IF\_DELIVERY\_ENTRY = ***, CID\_CWORD37\_IF\_TX\_START,  
CID\_CWORD37\_IF\_TX\_STOP, CID\_CWORD37\_IF\_CMD\_CTRL,  
CID\_CWORD37\_IF\_COMM\_WATCH, CID\_CWORD34\_IF\_DELIVERY\_ENTRY = ***,  
CID\_CWORD34\_IF\_SEND, CID\_CWORD34\_IF\_COMM\_WATCH,  
CID\_CWORD34\_INITCMP\_REQ, CID\_CWORD34\_IF\_ENTRY\_CLEAR\_FOR\_DEBUG,  
CID\_CWORD34\_IF\_COMMWATCH\_CLEAR\_FOR\_DEBUG,  
CID\_CWORD34\_IF\_SWITCH\_INITCMP\_FOR\_DEBUG }  
enum \_PS\_Communication\_Internal\_DebugProtocol { CID\_CANIF\_DELETE\_DELIVERY\_ENTRY = ***,  
CID\_CANIF\_COMMWATCH\_CLEAR\_FOR\_DEBUG }  
enum CAN\_CWORD30\_TYPE { CAN\_CWORD30\_TX\_START = ***,  
CAN\_CWORD30\_TX\_SEQ\_FIN }
```

Functions

[CANIF_RET_API Canif_DeliveryEntry](#) (HANDLE h_app, PCSTR notify_name, uint8_t can_num, [CANID](#) *p_can_id)

[CANIF_RET_API Canif_Diag_CWORD29_DeliveryEntry](#) (const HANDLE h_app, const PCSTR notify_name, const uint8_t opc_num, const uint16_t *const p_opc)

[CANIF_RET_API Canif_TransmissionStart](#) (HANDLE h_app, PCSTR notify_name, uint8_t rid, uint16_t freq, [CAN_DATA](#) *p_data)

[CANIF_RET_API Canif_TransmissionStop](#) (HANDLE h_app, PCSTR notify_name, [CANID](#) can_id)

[CANIF_RET_API Canif_CWORD30_Transmission](#) (HANDLE h_app, PCSTR notify_name, uint8_t rid, uint8_t err_type, uint8_t n_ta, [CANID](#) can_id, uint32_t data_size, uint8_t *p_data)

[CANIF_RET_API Canif_CWORD30_TxseqFinTransmission](#) (HANDLE h_app, PCSTR notify_name, uint8_t rid, uint8_t n_ta, [CANID](#) can_id, uint32_t data_size, uint8_t *p_data)

[CANIF_RET_API Canif_Diag_CWORD29_Transmission](#) (HANDLE h_app, PCSTR notify_name, const uint16_t opc, uint8_t rid, uint32_t data_size, uint8_t *p_data)

[CANIF_RET_API Canif_CommandCtrl](#) (HANDLE h_app, PCSTR notify_name, uint8_t rid, uint32_t cmd_id)

[CANIF_RET_API Canif_CommWatch](#) (HANDLE h_app, PCSTR notify_name, [CANID](#) can_id, [DID](#) did, uint16_t watch_time)

[CANIF_RET_API Canif_TransStart](#) (HANDLE h_app, [CANID](#) can_id, [CAN_DATA_MASK](#) *mask, [CAN_DATA_BIT](#) *dat, uint32_t freq)

[CANIF RET API Canif TransStop](#) (HANDLE h_app, [CANID](#) can_id)
[CANIF RET API Canif Send](#) ([CANID](#) can_id, [CAN DATA MASK](#) *mask, [CAN DATA BIT](#) *dat)
[CANIF RET API _CWORD37 if DeliveryEntry](#) (HANDLE h_app, PCSTR notify_name, uint8_t can_num, [CANID](#) *p_can_id)
[CANIF RET API _CWORD37 if TransmissionStart](#) (HANDLE h_app, PCSTR notify_name, uint16_t freq, uint8_t err_type, [_CWORD37 DATA](#) *p_data)
[CANIF RET API _CWORD37 if TransmissionStop](#) (HANDLE h_app, PCSTR notify_name, [CANID](#) can_id)
[CANIF RET API _CWORD37 if CommWatch](#) (HANDLE h_app, PCSTR notify_name, [CANID](#) can_id, [DID](#) did, uint16_t watch_time)
[CANIF RET API _CWORD37 if CommandCtrl](#) (HANDLE h_app, PCSTR notify_name, uint32_t cmd_id)
[CANIF _CWORD37 if ComposeCanID](#) (uint16_t uc_logic_id, uint8_t uc_send_id, uint8_t uc_rcv_id, uint16_t us_cmd_id)
[CANIF RET API _CWORD37 if AnalyzeCanID](#) ([CANID](#) ul_can_id, uint16_t *puc_logic_id, uint8_t *puc_send_id, uint8_t *puc_rcv_id, uint16_t *pus_cmd_id)
[CANIF RET API Canif Debug Delete AllDeliveryEntryList](#) (HANDLE h_app)
[CANIF RET API _CWORD34 DeliveryEntry](#) (HANDLE h_app, PCSTR notify_name, uint8_t can_num, [CANID](#) *p_can_id)
[CANIF RET API _CWORD34 Send](#) (HANDLE h_app, PCSTR notify_name, uint8_t send_id, uint8_t data_num, uint8_t *p_data)
[CANIF RET API _CWORD34 CommWatch](#) (HANDLE h_app, PCSTR notify_name, [CANID](#) can_id, [DID](#) did, uint16_t watch_time)

Detailed Description

Class Documentation

struct CAN_MSG_CANDATA_DAT

Data struct used to transfer CAN data (transfer data from CAN to user)

Class Members:

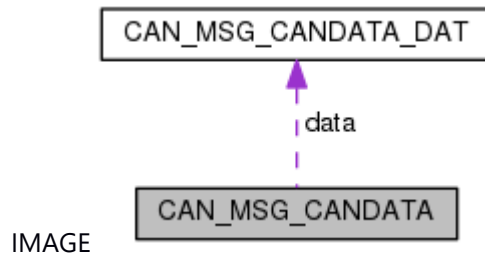
CANID	can_id	CAN ID
uint8_t	data[CAN DATA SIZE]	DATA[0]DATA[N]
uint8_t	dlc	Data Length
uint8_t	n_ta	N_TA
uint8_t	reserve[2]	reserve

uint8_t	reserve2	reserve

struct CAN_MSG_CANDATA

Data struct used to transfer CAN message (transfer message from CAN to user)

Collaboration diagram for CAN_MSG_CANDATA:



Class Members:

CAN MSG CANDATA DAT	data	data
T_APIMSG_MSGBUF_HEADER_COMM	hdr	header

struct CAN_MSG_CANGWDATA_DAT

Data struct used to transfer CAN data to CANGW (transfer data from CAN to user)

Class Members:

uint8_t	can_id_high	CAN ID upper byte
uint8_t	can_id_low	CAN ID lower byte
uint8_t	data[CANGW DLC MAX SIZE]	data
uint8_t	dlc	Data Length

struct CAN_MSG_CAN_CWORD29_DATA

Data struct used to transfer CAN CWORD29 data.(transfer data from CAN to user)

Class Members:

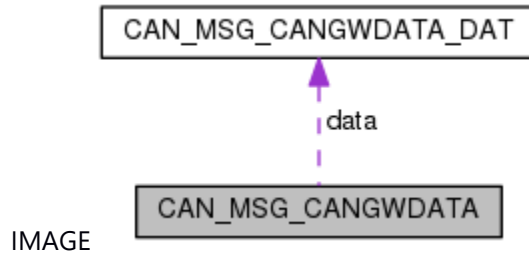
uint8_t	data[CAN_CWORD29_MEM_SZ]	Data buffer.
uint32_t	dlc	Data length(1~4100)

uint16_t	opc	OPC.
----------	-----	------

struct CAN_MSG_CANGWDATA

Message struct used to transfer CAN data to CANGW (transfer message from CAN to user)

Collaboration diagram for CAN_MSG_CANGWDATA:



Class Members:

CAN_MSG_CANGWDATA_DAT	data[CANGW_SND_NUM_MAX]	CAN data
uint8_t	num	Number of CAN data(max 22)

struct CAN_DATA

Data struct used to transfer CAN data (transfer data from user to CANIF API)

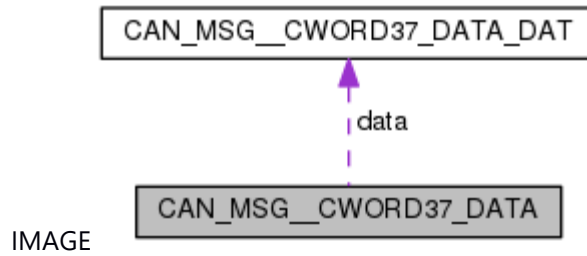
Class Members:

CANID	can_id	CAN ID
uint8_t	data[CAN_DATA_SIZE +1]	DATA
uint8_t	dlc	Data Length
uint8_t	reserve[3]	reserve

struct CAN_MSG_CWORD37_DATA

Message struct used to transfer CAN data (transfer message from CAN to user)

Collaboration diagram for CAN_MSG_CWORD37_DATA:



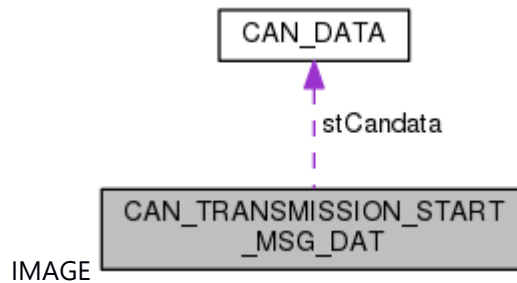
Class Members:

CAN_MSG_CWORD37_DATA_DAT	data	data
T_APIMSG_MSGBUF_HEADER_COMM	hdr	header

struct CAN_TRANSMISSION_START_MSG_DAT

Data struct used to transfer CAN data (transfer data from CANIF API to CAN)

Collaboration diagram for CAN_TRANSMISSION_START_MSG_DAT:



Class Members:

char	notifyName[MAX_NAME_SIZE_APP]	Delivery target thread name
CAN_DATA	stCandata	CAN DATA
uint8_t	ucRid	Resource ID
uint32_t	ulEventId	Thread ID used by <i>CWORD64</i>
uint16_t	usFreq	Cycle of send message(Unit of 100ms)

struct CAN_CWORD30_TRANS_START_MSG_DAT

Data struct used to transfer CAN data(*CWORD30*) (transfer data from CANIF API to CAN)

Class Members:

CAN_CWORD30_TYPE	_CWORD94_Type	Type of CAN data(CWORD30 transmission)
uint8_t	n_ta	Target address
char	notifyName[MAX_NAME_SIZE_APP]	Delivery target thread name
uint8_t	ucCsum	CSUM value
uint8_t	ucErrType	Error type
uint8_t	ucRid	Resource ID
CANID	ulCanId	CAN ID
uint32_t	ulDataSize	Data size
uint32_t	ulEventId	Event ID
uint16_t	usMyPid	PNO of caller thread

struct CAN_MSG_SENDSTS_DAT

Data struct used to notify transmission result (transfer data from CAN API to user)

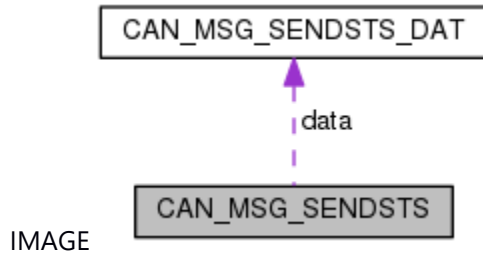
Class Members:

uint8_t	reserve[3]	Reserve
uint8_t	ucStatus	Transmission result status
CANID	ulCanid	CAN ID

struct CAN_MSG_SENDSTS

Message struct used to notify transmission result (transfer message from CAN API to user)

Collaboration diagram for CAN_MSG_SENDSTS:



Class Members:

CAN MSG SENDSTS DAT	data	Data
T_APIMSG_MSGBUF_HEADER_COMM	hdr	header

struct CAN_MSG_CWORD29_SENDSTS_DAT

Data struct used to notify *CWORD29* transmission result (transfer data from CAN API to user)

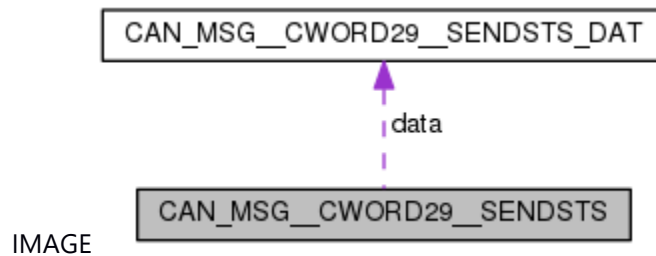
Class Members:

uint8_t	reserve[1]	Reserve
uint8_t	ucStatus	Transmission result status
uint16_t	usOpc	OPC

struct CAN_MSG_CWORD29_SENDSTS

Message struct used to notify *CWORD29* transmission result (transfer message from CAN API to user)

Collaboration diagram for CAN_MSG_CWORD29_SENDSTS:



Class Members:

CAN_MSG_CWORD29_SENDSTS_DAT	data	Data
T_APIMSG_MSGBUF_HEADER_COMM	hdr	header

struct CAN_FREQ_TRANS_STOP_MSG_DAT

Data struct used for regular transmission stop (transfer data from CANIF API to CAN)

Class Members:

char	notifyName[MAX_NAME_SIZE_APP]	Delivery target thread name
CANID	ulCanid	CAN ID
uint32_t	ulEventId	Event ID that use for CWORD64

struct CAN_MSG_COMM_WATCHSTS_DAT

Data struct used for communication stop/restore (transfer message from CAN to user)

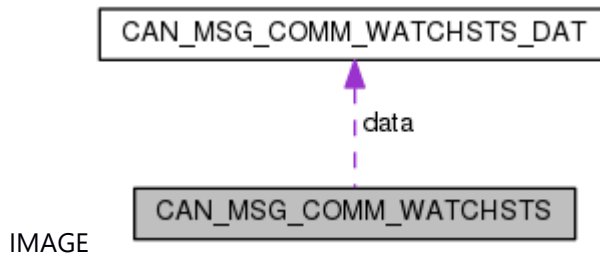
Class Members:

DID	ulDid	Data ID
---------------------	-------	---------

struct CAN_MSG_COMM_WATCHSTS

Data struct used for communication stop/restore (transfer message from CAN to user)

Collaboration diagram for CAN_MSG_COMM_WATCHSTS:



Class Members:

CAN_MSG_COMM_WATCHSTS_DAT	data	Data
T_APIMSG_MSGBUF_HEADER_COMM	hdr	Header

struct CAN_DELIVERY_ENTRY

Data struct used for delivery entry (transfer data from CANIF API to CAN)

Class Members:

char	notifyName[MAX_NAME_SIZE_APP]	Delivery target thread name
CANID	ulCanid[CAN_DELIVERY_CANID_ENTRY_MAX]	CAN ID entry array
uint32_t	ulEventId	Event ID use for <i>CWORD64</i>
uint16_t	usCanNum	Number of CAN ID entry

struct CAN_CWORD29_DELIVERY_ENTRY

_CWORD29_data struct used for delivery entry (transfer data from CANIF API to CAN)

Class Members:

char	notifyName[MAX_NAME_SIZE_APP]	Delivery target thread name
uint16_t	usOpc[CAN_DELIVERY_OPC_ENTRY_MAX]	OPC entry array
uint16_t	usOpcNum	Number of OPC entry

struct CAN_COMM_WATCH_MSG_DAT

Data struct used for communication stop (transfer data from CANIF API to CAN)

Class Members:

char	notifyName[MAX_NAME_SIZE_APP]	Delivery target thread name
CANID	ulCanid	CAN ID
DID	ulDid	Data ID
uint32_t	ulEvtId	Event ID
uint16_t	usWatchTime	Watch time for communication stop (Unit of 100ms)

struct CAN_CWORD30_ERRMSG_DAT

Data struct used for send/receive CAN data(CWORD30) error (transfer data from CAN API to user)

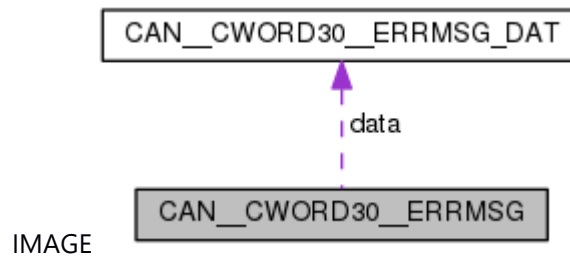
Class Members:

uint8_t	reserve[3]	Reserve
uint8_t	ucErrSt	Error status

struct CAN_CWORD30_ERRMSG

Message struct used for communication stop (transfer message from CANIF API to CAN)

Collaboration diagram for CAN_CWORD30_ERRMSG:

**Class Members:**

CAN_CWORD30_ERRMSG_DAT	data	Data
T_APIMSG_MSGBUF_HEADER_COMM	hdr	Header

struct CAN_CMD_CTRL_MSG_DAT

Data struct used for CAN command control (transfer data from CANIF API to CAN)

Class Members:

char	notifyName[MAX_NAME_SIZE_APP]	Delivery target thread name
uint8_t	ucCmdid	CAN command ID
uint8_t	ucRid	Resource ID
uint32_t	ulEvtId	Event ID

struct CAN_MSG_CANCMD_DAT

Data struct used for CAN command delivery (transfer data from CAN to user)

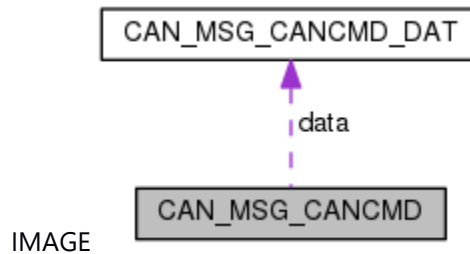
Class Members:

uint8_t	cmd_id	CAN command ID
uint8_t	data[CANCMD_DAT_MAX]	Data
uint8_t	reserve[3]	Reserve

struct CAN_MSG_CANCMD

Message struct used for CAN command delivery (transfer message from CAN to user)

Collaboration diagram for CAN_MSG_CANCMD:

**Class Members:**

CAN MSG CANCMD DAT	data	Data
T_APIMSG_MSGBUF_HEADER_COMM	hdr	Header

struct CAN_CMD_MSG_SENDSCTS_DAT

Data struct used for CAN command transmission result (transfer data from CAN to user)

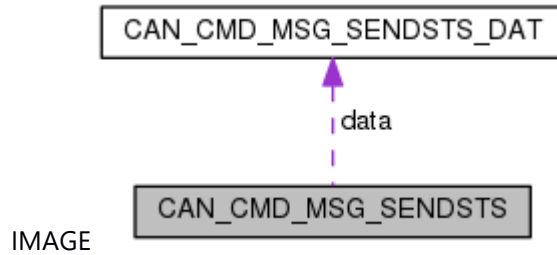
Class Members:

uint32_t	cmd_id	CAN command ID(same as delivery format)
uint8_t	reserve[3]	Reserve
uint8_t	status	Delivery result status

struct CAN_CMD_MSG_SENDSTS

Data struct used for CAN command transmission result (transfer data from CAN to user)

Collaboration diagram for CAN_CMD_MSG_SENDSTS:



Class Members:

CAN_CMD_MSG_SENDSTS_DAT	data	Data
T_APIMSG_MSGBUF_HEADER_COMM	hdr	Header

struct CAN_MSG_DATA

Data struct for receiving message(work data)

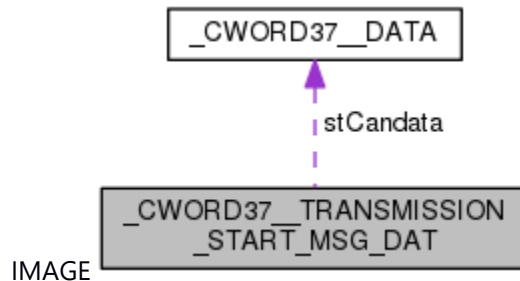
Class Members:

T_APIMSG_MSGBUF_HEADER_COMM	stHead	Header
uint8_t	ucData[CAN_MSGBUF_DATAMAX_SIZE]	Data

struct _CWORD37_TRANSMISSION_START_MSG_DAT

Data struct used for CAN data transmission start (transfer data from CANIF API to CAN)

Collaboration diagram for _CWORD37_TRANSMISSION_START_MSG_DAT:



Class Members:

char	notifyName[MAX_NAME_SIZE_APP]	Delivery target thread name
------	-------------------------------	-----------------------------

_CWORD37_DATA	stCandata	CAN data
uint8_t	ucerr	Error type
uint8_t	ucRid	Resource ID
uint32_t	ulEventId	Event ID
uint16_t	usFreq	Cycle of regular transmission (Unit of 100ms)

struct _CWORD37_MSG_DATA

Data struct for receiving message(work data)

Class Members:

T_APIMSG_MSGBUF_HEADER_COMM	stHead	Header
uint8_t	ucData[_CWORD37_MSGBUF_DATAMAX_SIZE]	Data

struct CAN_COMM_WATCHEXT_MSG_DAT

CAN data struct of communication stop registration (transfer data from CANIF API to CAN)

Class Members:

char	notifyName[MAX_NAME_SIZE_APP]	Delivery target thread name
uint8_t	uclg	has IG cooperation or not
CANID	ulCanid	CAN ID
DID	ulDid	Data ID
uint32_t	ulEvtId	Event ID for <i>CWORD64</i>
uint16_t	usWatchTime	Watch time for communication stop(Unit of

		100ms)
--	--	--------

struct CAN_DATA_MASK

Mask data struct for CAN data

Class Members:

uint8_t	dat[CAN_TXDATA_SIZE]	Mask data
---------	--	-----------

struct CAN_DATA_BIT

Data struct used for CAN data transmission

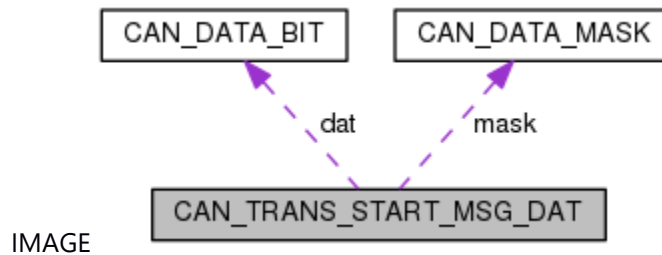
Class Members:

uint8_t	dat[CAN_TXDATA_SIZE]	Transmission data
---------	--	-------------------

struct CAN_TRANS_START_MSG_DAT

Data struct used for CAN data transmission registration

Collaboration diagram for CAN_TRANS_START_MSG_DAT:



Class Members:

CAN_DATA_BIT	dat	Transmission data
uint32_t	freq	Cycle of regular transmission
CANID	id	CAN ID
CAN_DATA_MASK	mask	Mask data

struct CAN_MSG_CWORD34_DATA

Struct used for CWORD2 data delivery

Class Members:

CANID	can_id	CAN ID
uint8_t	data[CWORD34_DLC_MAX_SIZE]	Data
uint8_t	dlc	Data Length
uint8_t	reserve[3]	Reserve

struct _CWORD34_MSG_SENDSTS

Struct used for *CWORD2* data send complete

Class Members:

CANID	can_id	CAN ID
uint8_t	reserve[2]	Reserve
uint8_t	send_id	Send ID
uint8_t	send_res	Send resource

struct _CWORD34_MSG_COMM_WATCHSTS

Message struct used for *CWORD2* communication stop/restore notification

Class Members:

DID	ulDid	Data ID
---------------------	-------	---------

Macro Definition Documentation

#define _CWORD34_DLC_MAX_SIZE 0x08

MAX length of *CWORD2* DLC

#define _CWORD34_SND_NUM_MAX 255

MAX number of *CWORD2* transmission data

#define _CWORD37_CANID_AREA_SIZE 4

CAN-ID area size

#define _CWORD37_CMDID_MASK 0x7FF

Bit mask of command ID

#define _CWORD37_CMDID_SHIFT 0

Bit shift of command ID

#define _CWORD37_DATAMAX_SIZE 2048

MAX CAN data size used to send message

#define _CWORD37_DLC_AREA_SIZE 2

DLC area size

#define _CWORD37_ERR_NONREQ 0

No notification

#define _CWORD37_ERROR_CANID 0xFFFFFFFF

Invalid CAN ID

#define _CWORD37_LOGICID_MASK 0x3FF

Bit mask of logic ID

#define _CWORD37_LOGICID_SHIFT 19

Bit shift of logic ID

#define _CWORD37_MSGBUF_DATAMAX_SIZE ([_CWORD37_MSGBUF_MAX_SIZE](#) - [CAN_MSGBUF_HEADER_SIZE](#))

MAX size of message buffer data

#define _CWORD37_MSGBUF_MAX_SIZE 2076

MAX size of message buffer

#define _CWORD37_RCVID_MASK 0x0F

Bit mask of receive ID

#define _CWORD37_RCVID_SHIFT 11

Bit shift of receive ID

#define _CWORD37_SENDID_MASK 0x0F

Bit mask of send ID

#define _CWORD37_SENDID_SHIFT 15

Bit shift of send ID

#define

_CWORD37_TRANSMISSION_SHM_NAME "/_CWORD37_TRANSMISSION_SHM_%d"

Shared memory name used by `_CWORD37_if_TransmissionStart`

#define _CWORD37_TXERR_REQ 1

Request of transmission error notification

#define CAN_CWORD29_MEM_SZ (4100)

Memory size used for `CWORD29` can data

#define CAN_CWORD30_ERR_NONREQ 0

No notification

#define CAN_CWORD30_N_TA_NONE (0x00)

`CWORD30` no target address

#define CAN_CWORD30_RXERR_REQ 1

Request of receive error notification

#define CAN_CWORD30_SEMAPHO_NAME ("CAN_CWORD30_SEMAPHO")

Semaphore name used to read *CWORD30* 's send data

#define CAN_CWORD30_SHARED_MEM_NAME ("CAN_CWORD30_SHARE_MEMORY")

Shared memory name used to read *CWORD30* 's send data

#define CAN_CWORD30_SHARED_MEM_SZ (1024)

Shared memory size used to read *CWORD30* 's send data

#define CAN_CWORD30_SHM_NAME "/CAN_CWORD30_SHM_%d"

Shared memory name used by Canif_CWORD30_Transmission

#define CAN_CWORD30_TXDATA_SIZE CAN_CWORD30_SHARED_MEM_SZ

Shared memory size used to read *CWORD30* 's transmission data

#define CAN_CWORD30_TXERR_REQ 2

Request of send error notification

#define CAN_BUFFERFUL 2

Send buffer full

#define CAN_CMDID_BUS_STATUS_REQ_TX 0x05

Request CAN BUS status notification ToDo()busl/F

#define CAN_CMDID_BUS_STATUS_RESP_RX 0x0C

Response of CAN bus status receive ToDo()busl/F

#define CAN_CMDID_CONNECTION_NODE_REQ_TX 0x04

Request CAN connected node notification

#define CAN_CMDID_CONNECTION_NODE_RESP_RX 0x0B

Response of CAN connection node receive

#define CAN_CMDID_FUELCALC_REQ_TX 0x06

Response of CAN fuel calculator reset

#define CAN_CMDID_FUELCALC_RST_REQ_DELIVERY 0x00

Request CAN fuel calculator reset

#define CAN_CMDID_FUELCALC_RST_REQ_RX 0x07
Receive CAN fuel calculator reset request

#define CAN_CMDID_MRST_INFO_REQ_TX 0x02
Request CAN Master reset notification

#define CAN_CMDID_MRST_INFO_RESP_RX 0x09
Receive CAN Master reset notification

#define CAN_CMDID_STARTUP_FIN_REQ_TX 0x01
Request CAN startup finished notification

#define CAN_CMDID_STARTUP_FIN_RESP_RX 0x08
Receive CAN startup finished notification

#define CAN_CMDID_VERSION_REQ_TX 0x03
Request CAN version

#define CAN_CMDID_VERSION_RESP_RX 0x0A
Response of CAN version receive

#define CAN_DATA_SIZE 63
MAX CAN data size used to send/receive message

#define CAN_DEBUG
Debug defination for DeliveryEntry IT Test

#define CAN_DELIVERY_RESTART 1
Delivery restart

#define CAN_DELIVERY_CANID_ENTRY_MAX 125
MAX number of delivery entry

#define CAN_DELIVERY_OPC_ENTRY_MAX 255
OPC MAX number of delivery entry

#define CAN_DELIVERY_STOP 0

Delivery stop

#define CAN_IG_COOPERATION_OFF 0

IG cooperation off

#define CAN_IG_COOPERATION_ON 1

IG cooperation on

#define CAN_MSGBUF_DATAMAX_SIZE ([CAN_MSGBUF_MAX_SIZE - CAN_MSGBUF_HEADER_SIZE](#))

MAX size of message buffer header

#define CAN_MSGBUF_HEADER_SIZE 16

Size of message buffer header

#define CAN_MSGBUF_MAX_SIZE 528

MAX size of message buffer

#define CAN_NTA_INVALID 0xFF

Invalid target address

#define CAN_RETRYOUT 1

Retryout

#define CAN_RID_NOTUSE_CODE 0xFF

The code of resource ID which is not used

#define CAN_SUCCESS 0

Success

#define CAN_TXDATA_SIZE 8

MAX CAN data size used to send message

#define CANCEMD_DAT_MAX 68

MAX size of CAN command data

#define CANGW_DLC_MAX_SIZE 0x08

MAX length of CANGW DLC

#define CANGW_SND_NUM_MAX 0x16

MAX number of CANGW transmission data

#define CANIF_RET_ERROR_BUFFULL 3

Buffer Full

#define CANIF_RET_ERROR_CANCEL 9

Error Cancel

#define CANIF_RET_ERROR_CANIDFULL 4

CAN ID FULL

#define CANIF_RET_ERROR_CREATE_EVENT 8

Event Creat Error

#define CANIF_RET_ERROR_NULL 10

Null Pointer

#define CANIF_RET_ERROR_PARAM 2

Parameter error

#define CANIF_RET_ERROR_PID 6

Thread ID unmatched

#define CANIF_RET_ERROR_TIMER 7

Timer error

#define CANIF_RET_ERROR_UNDEF 5

ID unregister

#define CANIF_RET_NORMAL 1

Sucess

#define LAN_SERVICE_CWORD34_ "_CWORD34_COM_PROT"

Defination of Service name

#define LAN_SERVICE_CWORD37_ "_CWORD37_COM_PROT"

Defination of Service name

#define LAN_SERVICE_CAN "CAN_COM_PROT"

Defination of Service name

#define

NTFY_Communication_CWORD34_ISAVAILABLE "Communication/_CWORD34_IsAvailable"

Availability notification of *CWORD34*

#define

NTFY_Communication_CWORD37_ISAVAILABLE "Communication/_CWORD37_IsAvailable"

Availability notification of *CWORD26*

#define NTFY_Communication_CAN_ISAVAILABLE "Communication/CanIsAvailable"

Availability notification of CAN

Typedef Documentation

typedef uint32_t [CANID](#)

define of CAN ID type

typedef int32_t [CANIF_RET_API](#)

CAN I/F API return code

typedef uint32_t [DID](#)

data ID

typedef enum [_PS_Communication_Internal_DebugProtocol](#) [_PS_Communication_Internal_DebugProtocol](#)

Defination of Command ID (Used for debug)

typedef enum [_PS_Communication_InternalProtocol](#) [_PS_Communication_InternalProtocol](#)

Defination of Command ID (CANIF -> CAN)

typedef enum [_PS_CommunicationProtocol](#) [_PS_CommunicationProtocol](#)

Defination of Command ID (CAN -> user)

Enumeration Type Documentation

enum [_PS_Communication_Internal_DebugProtocol](#)

Defination of Command ID (Used for debug)

Enumerator

CID_CANIF_DELETE_DELIVERY_ENTRY Delivery unregister (Used for debug)

CID_CANIF_COMMWATCH_CLEAR_FOR_DEBUG CANIF communication data all clear(Used for debug)

enum [_PS_Communication_InternalProtocol](#)

Defination of Command ID (CANIF -> CAN)

Enumerator

CID_CANIF_DELIVERY_ENTRY Delivery registration of CAN data

CID_CANIF_CWORD29_DELIVERY_ENTRY Delivery registration of
_CWORD29_data

CID_CANIF_TX_START CAN data transmission start
CID_CANIF_TX_STOP CAN data transmission stop
CID_CANIF_COMM_WATCH CAN data communication stop watching
CID_CANIF_CWORD30_TX_START CAN data(CWORD30) transmission start
CID_CANIF_CWORD30_TX_SEQ_FIN CAN data(CWORD30) transmission sequence complete
CID_CANIF_CWORD29_TX_START CAN data(CWORD29) transmission
CID_CANIF_CMD_CTRL CAN command control
CID_CANIF_TX_BIT_START CAN data transmission start
CID_CANIF_TX_BIT_STOP CAN data regular transmission stop
CID_CWORD37_IF_DELIVERY_ENTRY **CID_CWORD37_IF_TX_START**
CID_CWORD37_IF_TX_STOP **CID_CWORD37_IF_CMD_CTRL**
CID_CWORD37_IF_COMM_WATCH **CID_CWORD34_IF_DELIVERY_ENTRY**
Delivery registration CWORD2 data
CID_CWORD34_IF_SEND CWORD2 data transmission
CID_CWORD34_IF_COMM_WATCH CWORD2 data communication stop watching
CID_CWORD34_INITCMP_REQ The timer that use to CWORD2 initialize completion
CID_CWORD34_IF_ENTRY_CLEAR_FOR_DEBUG CWORD2 data all clear for debug
CID_CWORD34_IF_COMMWATCH_CLEAR_FOR_DEBUG CWORD2 communication data all clear(Used for debug)
CID_CWORD34_IF_SWITCH_INITCMP_FOR_DEBUG Change of CWORD2 startup finished flag for debug

enum [PS CommunicationProtocol](#)

Defination of Command ID (CAN -> user)

Enumerator

CID_CAN_DATA_DELIVERY Delivery CAN Data
CID_CAN_COMM_STOP CAN communication stop
CID_CAN_COMM_RESTORE CAN communication restore
CID_CAN_TX_RESULT CAN transmission result
CID_CAN_CWORD29_TX_RESULT CAN CWORD29 transmission result
CID_CAN_CWORD30_DELIVERY Delivery CAN data(CWORD30)
CID_CAN_DIAG_CWORD29_DELIVERY Delivery CAN data(CWORD29)
CID_CAN_CWORD30_FF_DELIVERY Delivery CAN data(CWORD30) FF

CID_CAN_CWORD30_RXERR CAN data(CWORD30) receive error
CID_CAN_CWORD30_TXERR CAN data(CWORD30) transmission error
CID_CAN_CMD_DELIVERY Delivery CAN command
CID_CAN_CMD_TX_RESULT CAN command transmission result
CID_CAN_CANGW_DELIVERY Delivery CAN data to CANGW
CID_CWORD37_DATA_DELIVERY **CID_CWORD37_CMD_DELIVERY**
CID_CWORD37_COMM_STOP **CID_CWORD37_COMM_RESTORE**
CID_CWORD37_TX_ERROR **CID_CWORD34_DATA_DELIVERY** Delivery
CWORD2 data
CID_CWORD34_COMM_STOP CWORD2 data communication stop
CID_CWORD34_COMM_RESTORE CWORD2 data communication restore
CID_CWORD34_SEND_RES Sending CWORD2 data complete

enum [CAN_CWORD30_TYPE](#)

Type of CAN data(CWORD30) transmission

Enumerator

CAN_CWORD30_TX_START Notification of CAN data(CWORD30)
transmission
CAN_CWORD30_TX_SEQ_FIN Notification of CAN data(CWORD30)
transmission complete

Function Documentation

[CANIF_RET_API_CWORD34_CommWatch](#) (HANDLE *h_app*, PCSTR *notify_name*, [CANID](#) *can_id*, [DID](#) *did*, uint16_t *watch_time*)

Brief

CWORD2 data regular transmission stopwatching

Parameters:

in	<i>h_app</i>	HANDLE - Handle for application
in	<i>notify_name</i>	PCSTR - Delivery target thread name
in	<i>can_id</i>	CANID - CAN ID
in	<i>did</i>	DID - Data ID
in	<i>watch_time</i>	uint16_t - Watch time

Return values:

<i>CANIF_RET_NORMAL</i>	Normality
<i>CANIF_RET_ERROR_PARAM</i>	Abnormality of parameter

<i>CANIF_RET_ERROR_CANCEL</i>	Abnormal termination
<i>CANIF_RET_ERROR_NULL</i>	NULL pointer

Prerequisite

None

Change of internal state

Change of internal state according to the API does not occur

Classification

Public

Type

Sync

See also:

None

[CANIF_RET_API_CWORD34_DeliveryEntry](#) (HANDLE *h_app*, PCSTR *notify_name*, uint8_t *can_num*, [CANID](#) * *p_can_id*)

Brief

Delivery registration of *CWORD2* data

Parameters:

in	<i>h_app</i>	HANDLE - Handle for application
in	<i>notify_name</i>	PCSTR - Delivery target thread name
in	<i>can_num</i>	uint8_t - Number of delivery entry
in	<i>p_can_id</i>	CANID* - Array of Destination IDs

Return values:

<i>CANIF_RET_NORMAL</i>	Normality
<i>CANIF_RET_ERROR_PARAM</i>	Abnormality of parameter
<i>CANIF_RET_ERROR_CANCEL</i>	Abnormal termination
<i>CANIF_RET_ERROR_NULL</i>	NULL pointer

Prerequisite

None

Change of internal state

Change of internal state according to the API does not occur

Classification

Public

Type

Sync

See also:

None

CANIF RET API `_CWORD34_Send` (HANDLE *h_app*, PCSTR *notify_name*, uint8_t *send_id*, uint8_t *data_num*, uint8_t* *p_data*)

Brief

CWORD2 data transmission starting

Parameters:

in	<i>h_app</i>	HANDLE - Handle for application
in	<i>notify_name</i>	PCSTR - Delivery target thread name
in	<i>send_id</i>	uint8_t - Send ID
in	<i>data_num</i>	uint8_t - Number of send CAN data
in	<i>p_data</i>	uint8_t* - Send CAN data

Return values:

<i>CANIF_RET_NORMAL</i>	Normality
<i>CANIF_RET_ERROR_PARAM</i>	Abnormality of parameter
<i>CANIF_RET_ERROR_CANCEL</i>	Abnormal termination
<i>CANIF_RET_ERROR_NULL</i>	NULL pointer

Prerequisite

None

Change of internal state

Change of internal state according to the API does not occur

Classification

Public

Type

ASync

See also:

None

CANIF RET API `_CWORD37_if_AnalyzeCanID` (CANID *ul_can_id*, uint16_t* *puc_logic_id*, uint8_t* *puc_send_id*, uint8_t* *puc_rcv_id*, uint16_t* *pus_cmd_id*)

Brief

Analyze CAN ID

Parameters:

in	<i>ul_can_id</i>	CANID - CAN ID
in	<i>puc_logic_id</i>	uint16_t* - Ponter of logic ID
in	<i>puc_send_id</i>	uint8_t* - Ponter of send ID
in	<i>puc_rcv_id</i>	uint8_t* - Ponter of receive ID
in	<i>pus_cmd_id</i>	uint16_t* - Ponter of command ID

Return values:

<i>CANIF_RET_NORMAL</i>	Normality
-------------------------	-----------

<code>CANIF_RET_ERROR_PARAM</code>	Abnormality of parameter
------------------------------------	--------------------------

Prerequisite

None

Change of internal state

Change of internal state according to the API does not occur

Classification

Public

Type

Sync

See also:

None

[CANIF_RET_API_CWORD37_if_CommandCtrl](#) (HANDLE *h_app*, PCSTR *notify_name*, uint32_t *cmd_id*)

Brief

command control

Parameters:

in	<i>h_app</i>	HANDLE - Handle for application
in	<i>notify_name</i>	PCSTR - Delivery target thread name
in	<i>cmd_id</i>	uint32_t - CAN command ID(32bit)

Return values:

<code>CANIF_RET_NORMAL</code>	Normality
<code>CANIF_RET_ERROR_PARAM</code>	Abnormality of parameter
<code>CANIF_RET_ERROR_CANCEL</code>	Abnormal termination

Prerequisite

None

Change of internal state

Change of internal state according to the API does not occur

Classification

Public

Type

ASync

See also:

None

CANIF_RET_API _CWORD37_if_CommWatch (HANDLE *h_app*, PCSTR *notify_name*, CANID *can_id*, DID *did*, uint16_t *watch_time*)

Brief

data regular transmission stop watching

Parameters:

in	<i>h_app</i>	HANDLE - Handle for application
in	<i>notify_name</i>	PCSTR - Delivery target thread name
in	<i>can_id</i>	CANID - CAN ID
in	<i>did</i>	DID - Data ID
in	<i>watch_time</i>	CANID - Communication watch suspension time(Unit of 100ms)

Return values:

<i>CANIF_RET_NORMAL</i>	Normality
<i>CANIF_RET_ERROR_PARAM</i>	Abnormality of parameter
<i>CANIF_RET_ERROR_CANCEL</i>	Abnormal termination

Prerequisite

None

Change of internal state

Change of internal state according to the API does not occur

Classification

Public

Type

Sync

See also:

None

CANID _CWORD37_if_ComposeCanID (uint16_t *uc_logic_id*, uint8_t *uc_send_id*, uint8_t *uc_rcv_id*, uint16_t *us_cmd_id*)

Brief

Combine CAN ID

Parameters:

in	<i>uc_logic_id</i>	uint16_t - Logic ID
in	<i>uc_send_id</i>	uint8_t - Send ID
in	<i>uc_rcv_id</i>	uint8_t - Receive ID
in	<i>us_cmd_id</i>	uint16_t - Command ID

Return values:

<i>_CWORD37_ERROR_CANID</i>	Invalid CAN ID(0xFFFFFFFF)
<i>CANID</i>	Valid CAN ID

Prerequisite

None

Change of internal state

Change of internal state according to the API does not occur

Classification

Public

Type

Sync

See also:

None

[CANIF_RET_API_CWORD37_if_DeliveryEntry](#) (HANDLE *h_app*, PCSTR *notify_name*, uint8_t *can_num*, [CANID](#) * *p_can_id*)

Brief

Delivery registration of CAN data

Parameters:

in	<i>h_app</i>	HANDLE - Handle for application
in	<i>notify_name</i>	PCSTR - Delivery target thread name
in	<i>can_num</i>	uint8_t - Number of delivery registration CAN ID
in	<i>p_can_id</i>	CANID* - Pointer of delivery registration CAN ID array

Return values:

<i>CANIF_RET_NORMAL</i>	Normality
<i>CANIF_RET_ERROR_PARAM</i>	Abnormality of parameter
<i>CANIF_RET_ERROR_CANCEL</i>	Abnormal termination

Prerequisite

None

Change of internal state

Change of internal state according to the API does not occur

Classification

Public

Type

Sync

See also:

None

[CANIF_RET_API_CWORD37_if_TransmissionStart](#) (HANDLE *h_app*, PCSTR *notify_name*, uint16_t *freq*, uint8_t *err_type*, [CWORD37_DATA](#) * *p_data*)

Brief

data transmission starting

Parameters:

in	<i>h_app</i>	HANDLE - Handle for application
in	<i>notify_name</i>	PCSTR - Delivery target thread name
in	<i>freq</i>	uint16_t - Cycle of regular transmission
in	<i>err_type</i>	uint8_t - Type of error notification
in	<i>p_data</i>	_CWORD37_DATA* - Pointer of transmission data

Return values:

<i>CANIF_RET_NORMAL</i>	Normality
<i>CANIF_RET_ERROR_PARAM</i>	Abnormality of parameter
<i>CANIF_RET_ERROR_CANCEL</i>	Abnormal termination

Prerequisite

None

Change of internal state

Change of internal state according to the API does not occur

Classification

Public

Type

ASync

See also:

None

[CANIF_RET_API_CWORD37_if_TransmissionStop](#) (HANDLE *h_app*, PCSTR *notify_name*, [CANID](#) *can_id*)

Brief

data regular transmission stop

Parameters:

in	<i>h_app</i>	HANDLE - Handle for application
in	<i>notify_name</i>	PCSTR - Delivery target thread name
in	<i>can_id</i>	CANID - CAN ID

Return values:

<i>CANIF_RET_NORMAL</i>	Normality
<i>CANIF_RET_ERROR_PARAM</i>	Abnormality of parameter
<i>CANIF_RET_ERROR_CANCEL</i>	Abnormal termination

Prerequisite

None

Change of internal state

Change of internal state according to the API does not occur

Classification

Public

Type

Sync

See also:

None

[CANIF_RET_API](#) Canif_CWORD30_Transmission (HANDLE *h_app*, PCSTR *notify_name*, uint8_t *rid*, uint8_t *err_type*, uint8_t *n_ta*, [CANID](#) *can_id*, uint32_t *data_size*, uint8_t * *p_data*)

Brief

Transmission of CAN data (CWORD30) transmission notification

Parameters:

in	<i>h_app</i>	HANDLE - Handle for application
in	<i>notify_name</i>	PCSTR - Delivery target thread name
in	<i>rid</i>	uint8_t - Resource ID
in	<i>err_type</i>	uint8_t - Type of error notification
in	<i>n_ta</i>	uint8_t - Target address
in	<i>can_id</i>	CANID - CAN ID
in	<i>data_size</i>	uint32_t - Data size
in	<i>p_data</i>	uint8_t* - Pointer of transmission data

Return values:

<i>CANIF_RET_NORMAL</i>	Normality
<i>CANIF_RET_ERROR_PARAM</i>	Abnormality of parameter
<i>CANIF_RET_ERROR_CANCEL</i>	Abnormal termination

Prerequisite

None

Change of internal state

Change of internal state according to the API does not occur

Classification

Public

Type

ASync

See also:

None

[CANIF_RET_API](#) Canif_CWORD30_TxseqFinTransmission (HANDLE *h_app*, PCSTR *notify_name*, uint8_t *rid*, uint8_t *n_ta*, [CANID](#) *can_id*, uint32_t *data_size*, uint8_t* *p_data*)

Brief

Transmission of CAN data (CWORD30) transmission sequence end notification

Parameters:

in	<i>h_app</i>	HANDLE - Handle for application
in	<i>notify_name</i>	PCSTR - Delivery target thread name
in	<i>rid</i>	uint8_t - Resource ID
in	<i>n_ta</i>	uint8_t - Target address
in	<i>can_id</i>	CANID - CAN ID
in	<i>data_size</i>	uint32_t - Data size
in	<i>p_data</i>	uint8_t* - Pointer of transmission data

Return values:

<i>CANIF_RET_NORMAL</i>	Normality
<i>CANIF_RET_ERROR_PARAM</i>	Abnormality of parameter
<i>CANIF_RET_ERROR_CANCEL</i>	Abnormal termination

Prerequisite

None

Change of internal state

Change of internal state according to the API does not occur

Classification

Public

Type

ASync

See also:

None

[CANIF_RET_API](#) Canif_CommandCtrl (HANDLE *h_app*, PCSTR *notify_name*, uint8_t *rid*, uint32_t *cmd_id*)

Brief

CAN command control

Parameters:

in	<i>h_app</i>	HANDLE - Handle for application
in	<i>notify_name</i>	PCSTR - Delivery target thread name
in	<i>rid</i>	uint8_t - Resource ID

in	<i>cmd_id</i>	uint32_t - CAN command ID(32bit)
----	---------------	----------------------------------

Return values:

<i>CANIF_RET_NORMAL</i>	Normality
<i>CANIF_RET_ERROR_PARAM</i>	Abnormality of parameter
<i>CANIF_RET_ERROR_CANCEL</i>	Abnormal termination

Prerequisite

None

Change of internal state

Change of internal state according to the API does not occur

Classification

Public

Type

ASync

See also:

None

CANIF_RET_API Canif_CommWatch (HANDLE *h_app*, PCSTR *notify_name*, CANID *can_id*, DID *did*, uint16_t *watch_time*)

Brief

CAN data regular transmission stop

Parameters:

in	<i>h_app</i>	HANDLE - Handle for application
in	<i>notify_name</i>	PCSTR - Delivery target thread name
in	<i>can_id</i>	CANID - CAN ID
in	<i>did</i>	DID - Data ID
in	<i>watch_time</i>	uint16_t - Communication watch suspension time(Unit of 100ms)

Return values:

<i>CANIF_RET_NORMAL</i>	Normality
<i>CANIF_RET_ERROR_PARAM</i>	Abnormality of parameter
<i>CANIF_RET_ERROR_CANCEL</i>	Abnormal termination

Prerequisite

None

Change of internal state

Change of internal state according to the API does not occur

Classification

Public

Type

Sync

See also:

None

CANIF RET API Canif_Debug_Delete_AllDeliveryEntryList (HANDLE *h_app*)**Brief**

Delete all delivery entry

Parameters:

in	<i>h_app</i>	HANDLE - Handle for application
----	--------------	---------------------------------

Return values:

<i>CANIF_RET_NORMAL</i>	Normality
<i>CANIF_RET_ERROR_PARAM</i>	Abnormality of parameter
<i>CANIF_RET_ERROR_CANCEL</i>	Abnormal termination

Prerequisite

None

Change of internal state

Change of internal state according to the API does not occur

Classification

Public

Type

Sync

See also:

None

CANIF RET API Canif_DeliveryEntry (HANDLE *h_app*, PCSTR *notify_name*, uint8_t *can_num*, CANID * *p_can_id*)**Brief**

Delivery registration of CAN data

Parameters:

in	<i>h_app</i>	HANDLE - Handle for application
in	<i>notify_name</i>	PCSTR - Delivery target thread name
in	<i>can_num</i>	uint8_t - Number of delivery registration CAN ID
in	<i>p_can_id</i>	CANID* - Pointer of delivery registration CAN ID array

Return values:

<i>CANIF_RET_NORMAL</i>	Normality
<i>CANIF_RET_ERROR_PARAM</i>	Abnormality of parameter

<code>CANIF_RET_ERROR_CANCEL</code>	Abnormal termination
-------------------------------------	----------------------

Prerequisite

None

Change of internal state

Change of internal state according to the API does not occur

Classification

Public

Type

Sync

See also:

None

[CANIF_RET_API](#) Canif_Diag_CWORD29_DeliveryEntry (const HANDLE *h_app*, const PCSTR *notify_name*, const uint8_t *opc_num*, const uint16_t *const *p_opc*)

Brief

Delivery registration of _CWORD29_data

Parameters:

in	<i>h_app</i>	const HANDLE - Handle for application
in	<i>notify_name</i>	const PCSTR - Delivery target thread name
in	<i>opc_num</i>	const uint8_t - Number of delivery registration OPC
in	<i>p_opc</i>	const uint16_t* const - Pointer of delivery registration OPC array

Return values:

<code>CANIF_RET_NORMAL</code>	Normality
<code>CANIF_RET_ERROR_PARAM</code>	Abnormality of parameter
<code>CANIF_RET_ERROR_CANCEL</code>	Abnormal termination

Prerequisite

None

Change of internal state

Change of internal state according to the API does not occur

Classification

Public

Type

Sync

See also:

None

[CANIF_RET_API](#) Canif_Diag_CWORD29_Transmission (HANDLE *h_app*, PCSTR *notify_name*, const uint16_t *opc*, uint8_t *rid*, uint32_t *data_size*, uint8_t * *p_data*)

Brief

Transmission of CAN data (CWORD29) transmission

Parameters:

in	<i>h_app</i>	HANDLE - Handle for application
in	<i>notify_name</i>	PCSTR - Delivery target thread name
in	<i>opc</i>	const uint16_t - OPC
in	<i>rid</i>	uint8_t - Resource ID
in	<i>data_size</i>	uint32_t - Data size
in	<i>p_data</i>	uint8_t* - Pointer of transmission data

Return values:

<i>CANIF_RET_NORMAL</i>	Normality
<i>CANIF_RET_ERROR_PARAM</i>	Abnormality of parameter
<i>CANIF_RET_ERROR_CANCEL</i>	Abnormal termination

Prerequisite

None

Change of internal state

Change of internal state according to the API does not occur

Classification

Public

Type

ASync

See also:

None

[CANIF_RET_API](#) Canif_Send ([CANID](#) *can_id*, [CAN_DATA_MASK](#) * *mask*, [CAN_DATA_BIT](#) * *dat*)

Brief

Transmission CAN Command(for OpeningMovie)

Parameters:

in	<i>can_id</i>	CANID - CAN ID
in	<i>mask</i>	CAN_DATA_MASK* - Mask Data
in	<i>dat</i>	CAN_DATA_BIT* - Transmission Data

Return values:

<i>CANIF_RET_NORMAL</i>	Normality
<i>CANIF_RET_ERROR_PARAM</i>	Abnormality of parameter
<i>CANIF_RET_ERROR_CANCEL</i>	Abnormal termination

Prerequisite

None

Change of internal state

Change of internal state according to the API does not occur

Classification

Public

Type

ASync

See also:

None

[CANIF_RET_API](#) Canif_TransmissionStart (HANDLE *h_app*, PCSTR *notify_name*, uint8_t *rid*, uint16_t *freq*, [CAN_DATA](#) * *p_data*)

Brief

CAN data transmission starting

Parameters:

in	<i>h_app</i>	HANDLE - Handle for application
in	<i>notify_name</i>	PCSTR - Delivery target thread name
in	<i>rid</i>	uint8_t - Resource ID for CAN data transmission result notification
in	<i>freq</i>	uint16_t - Cycle of regular transmission
in	<i>p_data</i>	CAN_DATA* - Pointer of transmission data

Return values:

<i>CANIF_RET_NORMAL</i>	Normality
<i>CANIF_RET_ERROR_PARAM</i>	Abnormality of parameter
<i>CANIF_RET_ERROR_CANCEL</i>	Abnormal termination

Prerequisite

None

Change of internal state

Change of internal state according to the API does not occur

Classification

Public

Type

ASync

See also:

None

[CANIF RET API](#) Canif_TransmissionStop (HANDLE *h_app*, PCSTR *notify_name*, [CANID](#) *can_id*)

Brief

CAN data regular transmission stop

Parameters:

in	<i>h_app</i>	HANDLE - Handle for application
in	<i>notify_name</i>	PCSTR - Delivery target thread name
in	<i>can_id</i>	CANID - CAN ID

Return values:

<i>CANIF_RET_NORMAL</i>	Normality
<i>CANIF_RET_ERROR_PARAM</i>	Abnormality of parameter
<i>CANIF_RET_ERROR_CANCEL</i>	Abnormal termination

Prerequisite

None

Change of internal state

Change of internal state according to the API does not occur

Classification

Public

Type

Sync

See also:

None

[CANIF RET API](#) Canif_TransStart (HANDLE *h_app*, [CANID](#) *can_id*, [CAN DATA MASK](#) * *mask*, [CAN DATA BIT](#) * *dat*, uint32_t *freq*)

Brief

Transmission CAN Command

Parameters:

in	<i>h_app</i>	HANDLE - Handle for application
in	<i>can_id</i>	CANID - CAN ID
in	<i>mask</i>	CAN_DATA_MASK* - Mask Data
in	<i>dat</i>	CAN_DATA_BIT* - Transmission Data
in	<i>freq</i>	uint32_t - Transmission Cycle

Return values:

<i>CANIF_RET_NORMAL</i>	Normality
<i>CANIF_RET_ERROR_PARAM</i>	Abnormality of parameter
<i>CANIF_RET_ERROR_CANCEL</i>	Abnormal termination

Prerequisite

None

Change of internal state

Change of internal state according to the API does not occur

Classification

Public

Type

ASync

See also:

None

CANIF_RET_API Canif_TransStop (HANDLE *h_app*, CANID *can_id*)**Brief**

Stop periodic transmission of CAN Command

Parameters:

in	<i>h_app</i>	HANDLE - Handle for application
in	<i>can_id</i>	CANID - CAN ID

Return values:

<i>CANIF_RET_NORMAL</i>	Normality
<i>CANIF_RET_ERROR_PARAM</i>	Abnormality of parameter
<i>CANIF_RET_ERROR_CANCEL</i>	Abnormal termination

Prerequisite

None

Change of internal state

Change of internal state according to the API does not occur

Classification

Public

Type

Sync

See also:

None

LANstruct [_CWORD87_Lan_Cmdhdr](#)struct [_CWORD87_Lan_Cmd](#)struct [LAN_SHARE_CD_MUSIC_STATUS_DATA](#)

```

struct LAN\_DEVICE\_DAT
struct LAN\_MSG\_REGIST\_DEVICE\_DAT
struct LAN\_MSG\_REGIST\_DEVICE
struct \_CWORD87\_Lan\_Msg\_Trans\_physiadr\_tag
struct LAN\_MSG\_DELIVERY\_ENTRY\_DAT
struct LAN\_MSG\_DELIVERY\_ENTRY
struct LAN\_MSG\_DELIVERY\_DEVICE\_DAT
struct LAN\_MSG\_DELIVERY\_DEVICE
struct LAN\_MSG\_DELIVERY\_AMP\_DAT
struct LAN\_MSG\_DELIVERY\_AMP
struct LAN\_SVR\_MSG\_DELIVERY\_ENTRY\_DAT
struct LAN\_SVR\_MSG\_DELIVERY\_ENTRY
struct LAN\_SVR\_REGIST\_COMPLETION\_NOTIFY
struct \_CWORD87\_LAN\_MSG\_DELIVERY
struct \_CWORD87\_LAN\_DELIVERY\_ENTRY\_DAT
struct \_CWORD87\_LAN\_MSG\_DELIVERY\_ENTRY\_DAT
struct \_CWORD87\_LAN\_MSG\_DELIVERY\_ENTRY
struct \_CWORD87\_LAN\_MSG\_DELIVERY\_DEVICE\_ENTRY\_DAT
struct \_CWORD87\_LAN\_MSG\_DELIVERY\_DEVICE\_ENTRY
struct \_CWORD87\_LAN\_MSG\_TRANS\_DAT
struct \_CWORD87\_LAN\_MSG\_TRANS

```

Macros

```

#define LAN\_SERVICE\_CWORD87\_LAN "_CWORD87_LAN"
#define LAN\_SERVICE\_LANSVR "LAN_SERVER"
#define
    NTFY\_Communication\_CWORD87\_LAN\_ISAVAILABLE "Communication/_CWORD87_LANIsAv
    ailable"
#define CID\_LANSVR\_BASE (0xAA00)
#define CID\_CWORD87\_LAN\_BASE (0xAB00)
#define \_CWORD87\_LAN\_TOPPHYSIADR\_SYSMC (0x0FFF)
#define LAN\_TOPPHYSIADR\_UNKNOWN (0xFFFF)
#define LAN\_RET\_NORMAL (0)
#define LAN\_RET\_ERROR\_CREATE\_EVENT (1)
#define LAN\_RET\_ERROR\_PARAM (2)
#define LAN\_RET\_ERROR\_UNDEF (3)
#define LAN\_RET\_ERROR\_PID (4)
#define LAN\_RET\_ERROR\_BUFFULL (5)
#define LAN\_RET\_ERROR\_LAN\_srvldFULL (6)
#define LAN\_RET\_ERROR\_TIMER (7)
#define LAN\_RET\_ERROR\_CANCEL (8)
#define \_CWORD87\_LAN\_WILDCARD (0xFF)
#define LGADR\_NAVI (0x01)
#define LGADR\_CWORD50 (0x0A)
#define OPC\_SLAVE\_INFO\_REQ (0x81)
#define OPC\_SLAVE\_INFO\_RCV (0x91)
#define \_CWORD87\_LAN\_Queue\_ID\_Base EV_Queue_ID_Base(_CWORD87_LAN_MID)
#define \_CWORD87\_LAN\_CMD\_MAX 252

```

```

#define CID\_LANSVRIF\_DELIVERY\_DEVICE\_ENTRY (CID\_LANSVR\_BASE|0x0000)
#define CID\_LANSVRIF\_DELIVERY\_AMP\_ENTRY (CID\_LANSVR\_BASE|0x0001)
#define CID\_LANSVR\_DEVICE\_REGIST\_START (CID\_LANSVR\_BASE|0x0002)
#define CID\_LANSVR\_DELIVERY\_DEVICE (CID\_LANSVR\_BASE|0x0006)
#define CID\_LANSVR\_DELIVERY\_AMP (CID\_LANSVR\_BASE|0x0007)
#define CWORD87\_LAN\_IF\_DELIVERY\_DATA\_MAX (50)
#define LAN\_REGIST\_DEVICE\_MAX (100)
#define CWORD87\_LAN\_API\_DELIVERY\_ENTRY (1)
#define CWORD87\_LAN\_API\_DELIVERY\_MSG\_ENTRY (2)
#define LAN\_SHARE\_CD\_MUSIC\_STATUS ("LAN_SHARE_CD_MUSIC_STATUS")
#define CD\_MUSIC\_STATUS\_DATA\_INFORMATION ("CD_M_STATUS_DATA_INFORMATION")
#define CD\_MUSIC\_STATUS\_DATA\_SIZE 20
#define CD\_MUSIC\_STATUS\_DISCSTS\_CDROM 0x98

```

Typedefs

```

typedef int32_t LAN\_RET\_API
typedef struct CWORD87\_Lan\_Cmd\_CWORD87\_Lan\_Cmd
typedef enum CWORD87\_LAN\_Protocol\_CWORD87\_LAN\_Protocol
typedef enum CWORD87\_LAN\_Debug\_Protocol\_CWORD87\_LAN\_Debug\_Protocol

```

Enumerations

```

enum { LAN\_COMTYPE\_INDIVIDUAL, LAN\_COMTYPE\_GROUP, LAN\_COMTYPE\_ALL }
enum { LAN\_AMPSTS\_OFF, LAN\_AMPSTS\_ON }
enum { LAN_OP_ERASE = ***, LAN_OP_DISP }
enum LAN\_GloupID { LAN\_GLOUPID\_FIRST, LAN\_GLOUPID\_SECOND, LAN\_GLOUPID\_THIRD,
LAN\_GLOUPID\_MAX }
enum LAN\_BlockID { LAN\_BLOCKID\_FIRST, LAN\_BLOCKID\_SECOND, LAN\_BLOCKID\_THIRD,
LAN\_BLOCKID\_FORTH, LAN\_BLOCKID\_FIFTH, LAN\_BLOCKID\_SIXTH, LAN\_BLOCKID\_MAX }
enum LAN\_FlowPri { LAN\_FLOWPRI\_FIRST, LAN\_FLOWPRI\_SECOND, LAN\_FLOWPRI\_MAX }
enum CWORD87\_LAN\_Protocol { CID\_CWORD87\_LAN\_CMD\_RCV =
(CID_CWORD87_LAN_BASE|0x0005), CID\_CWORD87\_LAN\_DEVICE\_IND =
(CID_CWORD87_LAN_BASE|0x0007), CID\_CWORD87\_LANIF\_TRANSMISSION =
(CID_CWORD87_LAN_BASE|0x0000), CID\_CWORD87\_LANIF\_STOPREGTRANS =
(CID_CWORD87_LAN_BASE|0x0002), CID\_CWORD87\_LANIF\_DELIVERY\_ENTRY =
(CID_CWORD87_LAN_BASE|0x0004), CID\_CWORD87\_LANIF\_DELIVERY\_DEVICE\_ENTRY =
(CID_CWORD87_LAN_BASE|0x0006), CID\_CWORD87\_LANIF\_LOOPBACK =
(CID_CWORD87_LAN_BASE|0x0008), CID\_CWORD87\_LANIF\_LOOPBACK\_STOPREGTRANS =
(CID_CWORD87_LAN_BASE|0x0009), CID\_CWORD87\_LANIF\_SENDDOPDISPNOTIFY =
(CID_CWORD87_LAN_BASE|0x000A) }
enum CWORD87\_LAN\_Debug\_Protocol { CID\_CWORD87\_LANIF\_TRANS\_CLEAR\_DEBUG =
(CID_CWORD87_LAN_BASE|0x0010), CID\_CWORD87\_LANIF\_DELIVERY\_CLEAR\_DEBUG =
(CID_CWORD87_LAN_BASE|0x0011), CID\_CWORD87\_LANIF\_DEVICE\_CLEAR\_DEBUG =
(CID_CWORD87_LAN_BASE|0x0012) }

```

Functions

```

int32_t CWORD87\_Lan\_Transmission\_addPhysiAdr (HANDLE h_app, uint8_t com_type, uint16_t
physical_adr, CWORD87\_Lan\_Cmd *cmd, uint16_t cycle)

```


int32_t [_CWORD87_Lan_StopRegularTransmission](#) (HANDLE h_app, [_CWORD87_Lan_Cmdhdr](#) *cmd)

int32_t [_CWORD87_Lan_Transmission](#) (HANDLE h_app, uint8_t com_type, [_CWORD87_Lan_Cmd](#) *cmd, uint16_t cycle)

int32_t [_CWORD87_Lan_DeliveryMsgEntry](#) (HANDLE h_app, PCSTR notify_name, uint8_t datanum, [_CWORD87_Lan_Cmdhdr](#) *cmd_hdr)

int32_t [_CWORD87_Lan_DeliveryDeviceEntry](#) (HANDLE h_app, PCSTR notify_name)

int32_t [_CWORD87_Lan_GetPhysicalAddress](#) (uint8_t lgadr, uint16_t *phadr)

int32_t [_CWORD87_Lan_Loopback](#) (HANDLE h_app, uint8_t com_type, [_CWORD87_Lan_Cmd](#) *cmd, uint16_t cycle)

int32_t [_CWORD87_Lan_LoopbackStop](#) (HANDLE h_app, [_CWORD87_Lan_Cmdhdr](#) *cmd)

int32_t [_CWORD87_Lan_SendOpeningDispNotification](#) (uint8_t opDisp)

Detailed Description

Class Documentation

struct [_CWORD87_Lan_Cmdhdr](#)

CWORD87 -LAN command header structure

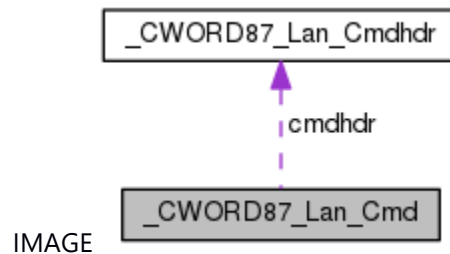
Class Members:

uint8_t	lgadr_from	Source logical address
uint8_t	lgadr_to	Forwarding destination logical address
uint8_t	opc	OP code
uint8_t	reserve	Reserve

struct [_CWORD87_Lan_Cmd](#)

CWORD87 -LAN command body structure

Collaboration diagram for [_CWORD87_Lan_Cmd](#):



Class Members:

_CWORD87_Lan_Cmdhdr	cmdhdr	LAN command header
uint8_t	data[_CWORD87_LAN_CMD_MAX]	Command data
uint16_t	data_length	Data length
uint16_t	reserve	Reserve

struct LAN_SHARE_CD_MUSIC_STATUS_DATA

Music status notification structure

Class Members:

uint8_t	ch_err_state	CH Status, Error Status
uint8_t	data10	DATA,Negative sign,DISC kind,ALL L/E,LOAD state,L/E Preparation,No recording,DTSTOP
uint8_t	data9	COMP,SCAN,D.SCAN,RPT,D.RPT,RAND, D.RAND, Compress Audio
uint8_t	disc_no	DISC NO
uint8_t	disk_info	Disk information
uint8_t	minute	Minute
uint8_t	mode_state	Mode status
uint8_t	reading_disc	READING DISC,Reserved,Play direction,Reserved
uint8_t	reserve[8]	Reserve
uint8_t	second	Second

uint8_t	track_end_minute	TRACK ending time minute
uint8_t	track_end_second	TRACK ending time second
uint8_t	track_no	TRACK NO

struct LAN_DEVICE_DAT

Connection equipment data part structure

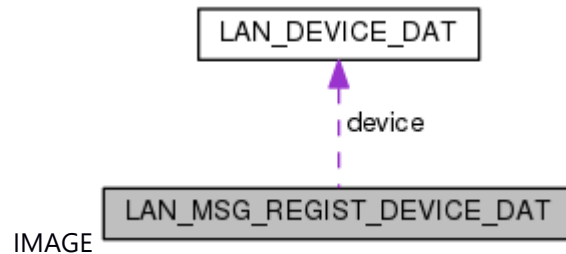
Class Members:

uint8_t	lgadr	Logical address
uint16_t	phadr	Physical address
uint8_t	rsv	Reserve

struct LAN_MSG_REGIST_DEVICE_DAT

Data part structure for connection equipment delivery

Collaboration diagram for LAN_MSG_REGIST_DEVICE_DAT:



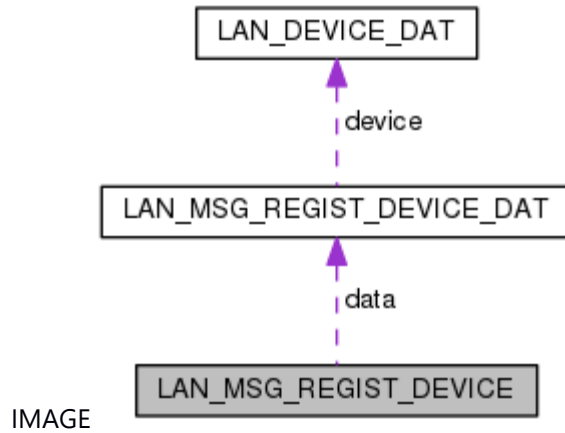
Class Members:

LAN_DEVICE_DAT	device[LAN_REGIST_DEVICE_MAX]	Connection device information
uint16_t	device_num	Number of connected devices
uint16_t	rsv	Reserve

struct LAN_MSG_REGIST_DEVICE

Message structure for connection device delivery notification

Collaboration diagram for LAN_MSG_REGIST_DEVICE:



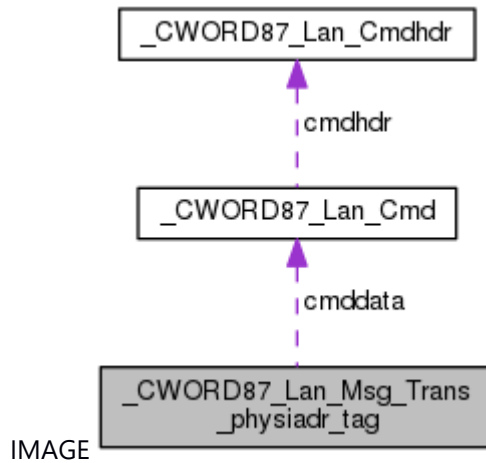
Class Members:

LAN MSG REGIST DEVICE DAT	data	Data section
T_APIMSG_MSGBUF_HEADER_COMM	hdr	Header part

struct _CWORD87_Lan_Msg_Trans_physiadr_tag

CWORD87 -LAN command Physical addressing transmisson structure

Collaboration diagram for _CWORD87_Lan_Msg_Trans_physiadr_tag:



Class Members:

_CWORD87_Lan_Cmd	cmddata	CWORD87 -LAN command body
uint8_t	com_type	Communication type
uint16_t	cycle	Periodical transmission cycle

uint16_t	physical_adr	Destination physical address

struct LAN_MSG_DELIVERY_ENTRY_DAT

Data part structure for delivery registration instruction

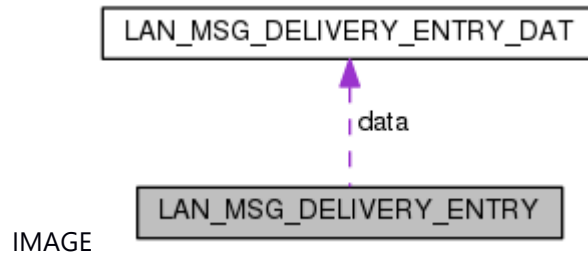
Class Members:

char	notifyName[MAX_NAME_SIZE_APP]	Notify Name
------	-------------------------------	-------------

struct LAN_MSG_DELIVERY_ENTRY

Message structure for delivery registration instruction notice

Collaboration diagram for LAN_MSG_DELIVERY_ENTRY:



Class Members:

LAN_MSG_DELIVERY_ENTRY_DAT	data	Data section
T_APIMSG_MSGBUF_HEADER_COMM	hdr	Header part

struct LAN_MSG_DELIVERY_DEVICE_DAT

Message structure for connection device delivery notification(Data section)

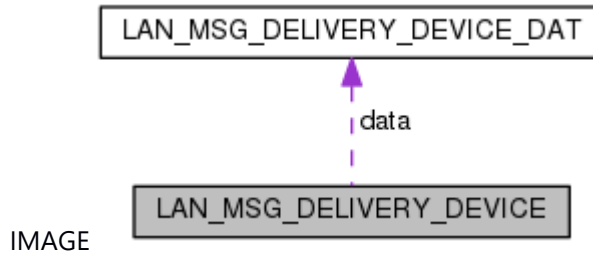
Class Members:

uint16_t	device[100]	Data section, top 1 byte[CWORD87 - LAN(0X01)/_CWORD39_(0x02)], lower 1 byte[Function address/Logical address]
uint16_t	device_num	Number of connected devices

struct LAN_MSG_DELIVERY_DEVICE

Message structure for connection device delivery notification

Collaboration diagram for LAN_MSG_DELIVERY_DEVICE:



Class Members:

LAN_MSG_DELIVERY_DEVICE_DAT	data	Connected device information
T_APIMSG_MSGBUF_HEADER_COMM	hdr	Header part

struct LAN_MSG_DELIVERY_AMP_DAT

Amp Power status delivery data section structure

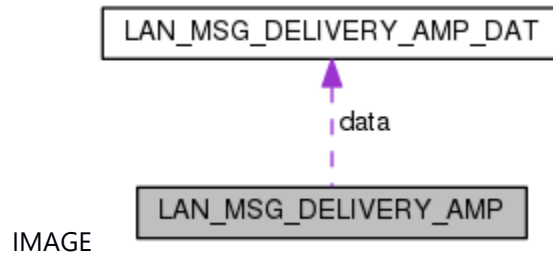
Class Members:

uint8_t	onOff	Amp Power supply state
---------	-------	------------------------

struct LAN_MSG_DELIVERY_AMP

Amp Power status delivery message structure

Collaboration diagram for LAN_MSG_DELIVERY_AMP:



Class Members:

LAN_MSG_DELIVERY_AMP_DAT	data	Data section
T_APIMSG_MSGBUF_HEADER_COMM	hdr	Header part

struct LAN_SVR_MSG_DELIVERY_ENTRY_DAT

Data part structure for delivery registration instruction

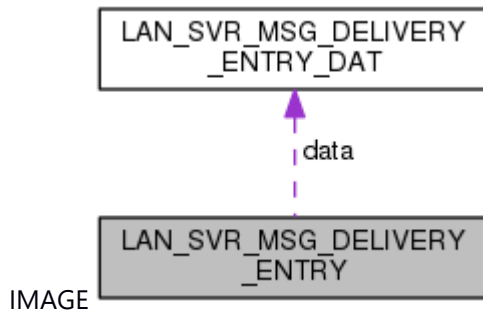
Class Members:

uint8_t	listNum	Total number of destinations
uint16_t	pNo	Process No

struct LAN_SVR_MSG_DELIVERY_ENTRY

Message structure for delivery registration instruction notice

Collaboration diagram for LAN_SVR_MSG_DELIVERY_ENTRY:



Class Members:

LAN_SVR_MSG_DELIVERY_ENTRY_DAT	data	Data section
T_APIMSG_MSGBUF_HEADER_COMM	hdr	Header part

struct LAN_SVR_REGIST_COMPLETION_NOTIFY

Message structure for registration completion notice

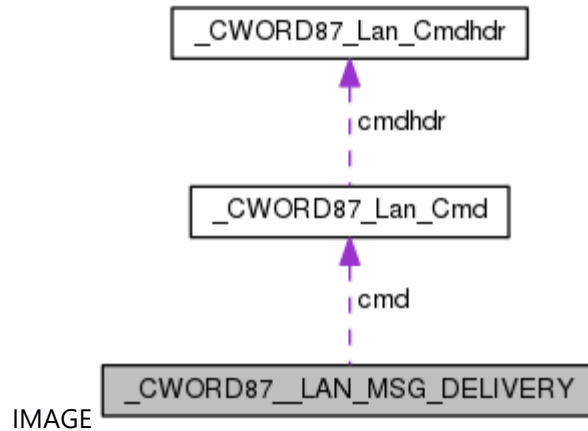
Class Members:

T_APIMSG_MSGBUF_HEADER_COMM	hdr	Header part
char	sender[MAX_NAME_SIZE_APP]	The sender name

struct _CWORD87_LAN_MSG_DELIVERY

Delivery message structure

Collaboration diagram for _CWORD87_LAN_MSG_DELIVERY:



Class Members:

_CWORD87_Lan_Cmd	cmd	Data section
T_APIMSG_MSGBUF_HEADER_COMM	hdr	Header part
uint16_t	phyadr_from	Physical address
uint8_t	reserve	Reserve
uint8_t	service_type	Communication type

struct _CWORD87__LAN_DELIVERY_ENTRY_DAT

Row data part structure for delivery registration instruction

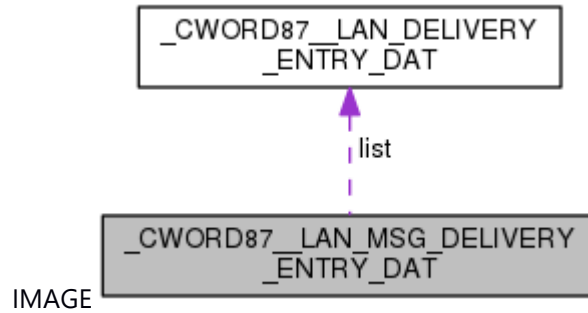
Class Members:

uint8_t	lgadr_from	Source logical address
uint8_t	lgadr_to	Forwarding destination logical address
char	notifyName[MAX_NAME_SIZE_APP]	Notify Name
uint8_t	opc	Op code

struct _CWORD87__LAN_MSG_DELIVERY_ENTRY_DAT

Data part structure for delivery registration instruction

Collaboration diagram for `_CWORD87_LAN_MSG_DELIVERY_ENTRY_DAT`:



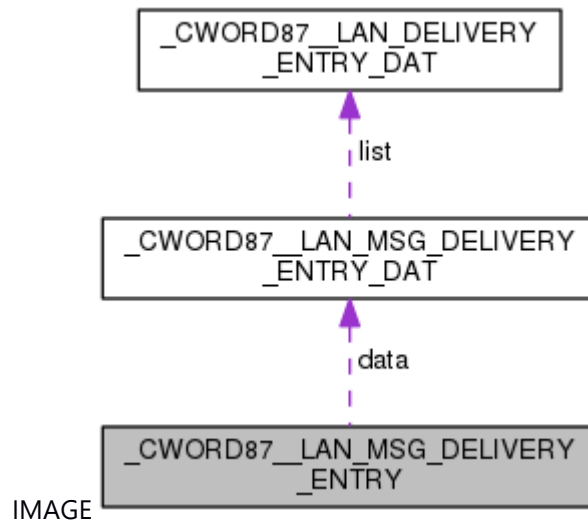
Class Members:

uint16_t	apiVersion	Api version
uint16_t	dataNum	Data Number
_CWORD87_LAN_DELIVERY_ENTRY_DAT	list[_CWORD87_LAN_IF_DELIVERY_DATA_MAX]	The list of data part structure for delivery registration instruction

struct `_CWORD87_LAN_MSG_DELIVERY_ENTRY`

Message structure for delivery registration notice

Collaboration diagram for `_CWORD87_LAN_MSG_DELIVERY_ENTRY`:



Class Members:

_CWORD87_LAN_MSG_DELIVERY_ENTRY_DAT	data	Data section
---	------	--------------

T_APIMSG_MSGBUF_HEADER_COMM	hdr	Header part
-----------------------------	-----	-------------

struct _CWORD87_LAN_MSG_DELIVERY_DEVICE_ENTRY_DAT

Device configuration delivery registration API request data section

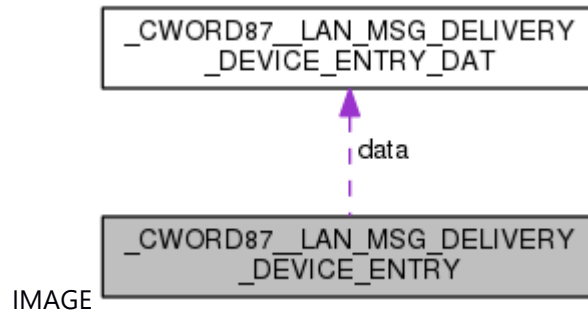
Class Members:

char	notifyName[MAX_NAME_SIZE_APP]	Notify Name
------	-------------------------------	-------------

struct _CWORD87_LAN_MSG_DELIVERY_DEVICE_ENTRY

Device configuration delivery registration API request

Collaboration diagram for _CWORD87_LAN_MSG_DELIVERY_DEVICE_ENTRY:



Class Members:

_CWORD87_LAN_MSG_DELIVERY_DEVICE_ENTRY_DAT	data	Data section
T_APIMSG_MSGBUF_HEADER_COMM	hdr	Header part

struct _CWORD87_LAN_MSG_TRANS_DAT

Transimission data part structure

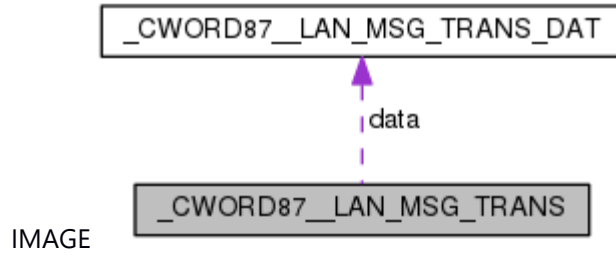
Class Members:

uint8_t	reserve	Reserve
---------	---------	---------

struct _CWORD87_LAN_MSG_TRANS

Message structure for delivery registration instruction notice

Collaboration diagram for _CWORD87_LAN_MSG_TRANS:



Class Members:

_CWORD87__LAN_MSG_TRANS_DAT	data	Data section
T_APIMSG_MSGBUF_HEADER_COMM	hdr	Header part

Macro Definition Documentation

#define _CWORD87_LAN_IF_DELIVERY_DATA_MAX (50)

IF delivery registration maximum number

#define _CWORD87_LAN_WILDCARD (0xFF)

Wildcard

#define _CWORD87_LAN_API_DELIVERY_ENTRY (1)

Old API version

#define _CWORD87_LAN_API_DELIVERY_MSG_ENTRY (2)

GPF API version

#define _CWORD87_LAN_CMD_MAX 252

CWORD87 -LAN command size maximum length

#define _CWORD87_LAN_Queue_ID_Base EV_Queue_ID_Base(_CWORD87_LAN_MID)

For ICR -> *CWORD87* LAN

#define _CWORD87_LAN_TOPHYSIADR_SYSMC (0x0FFF)

Physics address:SYS Micon

#define

CD_MUSIC_STATUS_DATA_INFORMATION ("CD_M_STATUS_DATA_INFORMATION")

Shared memory name

#define CD_MUSIC_STATUS_DATA_SIZE 20

Shared memory size

#define CD_MUSIC_STATUS_DISCSTS_CDROM 0x98

Shared memory size

#define CID_CWORD87_LAN_BASE (0xAB00)

CWORD87 -LAN

#define CID_LANSVR_BASE (0xAA00)

LAN-SVR

#define CID_LANSVR_DELIVERY_AMP (CID_LANSVR_BASE|0x0007)

LAN-SVR Amp power status delivery notification

#define CID_LANSVR_DELIVERY_DEVICE (CID_LANSVR_BASE|0x0006)

LAN-SVR connection equipment delivery notification

#define CID_LANSVR_DEVICE_REGIST_START (CID_LANSVR_BASE|0x0002)

LAN-SVR registration completion notice

#define CID_LANSVRIF_DELIVERY_AMP_ENTRY (CID_LANSVR_BASE|0x0001)

LAN-SVR Amp power status registration

#define CID_LANSVRIF_DELIVERY_DEVICE_ENTRY (CID_LANSVR_BASE|0x0000)

LAN-SVR register connected equipment

#define LAN_REGIST_DEVICE_MAX (100)

Device registration notification maximum number

#define LAN_RET_ERROR_BUFFULL (5)

Registration number is full(delivery/regular transmission/communication disruption monitoring)

#define LAN_RET_ERROR_CANCEL (8)

Abnormal termination(data destruction/cancellation)

#define LAN_RET_ERROR_CREATE_EVENT (1)

Create event failed

#define LAN_RET_ERROR_LAN_srvldFULL (6)

Registration number is full

#define LAN_RET_ERROR_PARAM (2)

The specified parameter is abnormal

#define LAN_RET_ERROR_PID (4)

Thread ID mismatch

#define LAN_RET_ERROR_TIMER (7)

Timer acquisition failed

#define LAN_RET_ERROR_UNDEF (3)

Unregistered ID

#define LAN_RET_NORMAL (0)

Normal termination

#define LAN_SERVICE_CWORD87_LAN "_CWORD87_LAN"

Service name

#define LAN_SERVICE_LANSVR "LAN_SERVER"

Service name

#define LAN_SHARE_CD_MUSIC_STATUS ("LAN_SHARE_CD_MUSIC_STATUS")

Semaphore name

#define LAN_TOPHYSIADR_UNKNOWN (0xFFFF)

Physics address:uncertain value

#define LGADR_CWORD50_ (0x0A)

Logical address *CWORD39* communication

#define LGADR_NAVI (0x01)

Logical address NAVI Micon management unit

#define

NTFY_Communication_CWORD87_LAN_ISAVAILABLE "Communication/_CWORD87_LANIs Available"

Availability notification of *CWORD87* -LAN

#define OPC_SLAVE_INFO_RCV (0x91)

Slave system information response

#define OPC_SLAVE_INFO_REQ (0x81)

Slave system information request

Typedef Documentation

typedef struct [_CWORD87 Lan Cmd](#) [_CWORD87 Lan Cmd](#)

CWORD87 -LAN command body structure

typedef enum [_CWORD87 LAN Debug Protocol](#) [_CWORD87 LAN Debug Protocol](#)

CID for *CWORD87* -LAN debug

typedef enum [_CWORD87 LAN Protocol](#) [_CWORD87 LAN Protocol](#)

CID for *CWORD87* -LAN

typedef int32_t [LAN_RET_API](#)

Return value define for LAN

Enumeration Type Documentation

anonymous enum

Communication type

Enumerator

LAN_COMTYPE_INDIVIDUAL 0x00:Individual

LAN_COMTYPE_GROUP 0x01:Group broadcast

LAN_COMTYPE_ALL 0x02:Simultaneous broadcast

anonymous enum

Amplifier power supply status

Enumerator

LAN_AMPSTS_OFF Amplifier power off

LAN_AMPSTS_ON Amplifier power off

enum [_CWORD87 LAN Debug Protocol](#)

CID for *CWORD87* -LAN debug

Enumerator

CID_CWORD87_LANIF_TRANS_CLEAR_DEBUG CWORD87 -LAN command periodic transmission all deletion(For debug)

CID_CWORD87_LANIF_DELIVERY_CLEAR_DEBUG Delete all CWORD87 -LAN command delivery registration(For debug)

CID_CWORD87_LANIF_DEVICE_CLEAR_DEBUG Delete all CWORD87 -LAN device delivery registration(For debug)

enum [CWORD87 LAN Protocol](#)

CID for CWORD87 -LAN

Enumerator

CID_CWORD87_LAN_CMD_RCV CWORD87 -LAN command delivery notification

CID_CWORD87_LAN_DEVICE_IND CWORD87 -LAN device configuration notification

CID_CWORD87_LANIF_TRANSMISSION CWORD87 -LAN command transmission request

CID_CWORD87_LANIF_STOPREGTRANS CWORD87 -LAN command stop periodical transmission

CID_CWORD87_LANIF_DELIVERY_ENTRY Delivery registration instruction

CID_CWORD87_LANIF_DELIVERY_DEVICE_ENTRY Device configuration delivery registration API request

CID_CWORD87_LANIF_LOOPBACK CWORD87 -LAN command loopback transmission request

CID_CWORD87_LANIF_LOOPBACK_STOPREGTRANS CWORD87 -LAN command stop loopback periodical transmission

enum [LAN BlockID](#)

Block ID for flow control

Enumerator

LAN_BLOCKID_FIRST First

LAN_BLOCKID_SECOND Second

LAN_BLOCKID_THIRD Third

LAN_BLOCKID_FORTH Fourth

LAN_BLOCKID_FIFTH Fifth

LAN_BLOCKID_SIXTH Sixth

LAN_BLOCKID_MAX Max

enum [LAN FlowPri](#)

Priority for flow control

Enumerator

- LAN_FLOWPRI_FIRST** First
- LAN_FLOWPRI_SECOND** Second
- LAN_FLOWPRI_MAX** Max

enum [LAN GloupID](#)

Group ID for flow control

Enumerator

- LAN_GLOUPID_FIRST** First
- LAN_GLOUPID_SECOND** Second
- LAN_GLOUPID_THIRD** Third
- LAN_GLOUPID_MAX** Max

Function Documentation

int32_t _CWORD87_Lan_DeliveryDeviceEntry (HANDLE *h_app*, PCSTR *notify_name*)

Brief

Delivery registration of *CWORD87* -LAN Device info

Parameters:

in	<i>h_app</i>	HANDLE - Handle for Application
in	<i>notify_name</i>	PCSTR - Registered thread name

Return values:

<i>LAN_RET_NORMAL</i>	: Success
<i>LAN_RET_ERROR_PARAM</i>	: Paramter error
<i>LAN_RET_ERROR_CANCEL</i>	: Send the message failed

Prerequisite

Availability of Communication is TRUE

Change of internal state

Change of internal state according to the API does not occur.

Classification

Public

Type

Async

See also:

None

int32_t _CWORD87_Lan_DeliveryMsgEntry (HANDLE *h_app*, PCSTR *notify_name*, uint8_t *datanum*, [_CWORD87_Lan_Cmdhdr](#) * *cmd_hdr*)

BriefDelivery registration of *CWORD87* -LAN data**Parameters:**

in	<i>h_app</i>	HANDLE - Handle for Application
in	<i>notify_name</i>	PCSTR - Registered thread name
in	<i>datanum</i>	uint8_t - Number of <i>cmd_hdr</i> []
in	<i>cmd_hdr</i>	_CWORD87_Lan_Cmdhdr * - Delivery filter

Return values:

<i>LAN_RET_NORMAL</i>	: Success
<i>LAN_RET_ERROR_PARAM</i>	: Paramter error
<i>LAN_RET_ERROR_CANCEL</i>	: Send the message failed

Prerequisite

Availability of Communication is TRUE

Change of internal state

Change of internal state according to the API does not occur.

Classification

Public

Type

Async

See also:

None

int32_t _CWORD87_Lan_GetPhysicalAddress (uint8_t *lgadr*, uint16_t * *phadr*)

Brief

Get physical address by logical address

Parameters:

in	<i>lgadr</i>	uint8_t - Logical address
in	<i>phadr</i>	uint16_t* - Pointer of result

Return values:

<i>LAN_RET_NORMAL</i>	: Success
<i>LAN_RET_ERROR_PARAM</i>	: Paramter error
<i>LAN_RET_ERROR_CANCEL</i>	: Failed

Prerequisite

Availability of Communication is TRUE

Change of internal state

Change of internal state according to the API does not occur.

Classification

Public

Type

Sync

See also:

None

**int32_t _CWORD87_Lan_Loopback (HANDLE *h_app*, uint8_t *com_type*, [_CWORD87_Lan_Cmd](#)
* *cmd*, uint16_t *cycle*)**

Brief

CWORD87 -LAN command loopback transmission request

Parameters:

in	<i>h_app</i>	HANDLE - Handle for Application
in	<i>com_type</i>	uint8_t - Communication type (LAN_COMTYPE_XXX)
in	<i>cmd</i>	_CWORD87_Lan_Cmd* - Command data
in	<i>cycle</i>	uint16_t - Periodic transmission cycle

Return values:

<i>LAN_RET_NORMAL</i>	: Success
<i>LAN_RET_ERROR_PARAM</i>	: Paramter error
<i>LAN_RET_ERROR_CANCEL</i>	: Send the message failed

Prerequisite

Availability of Communication is TRUE

Change of internal state

Change of internal state according to the API does not occur.

Classification

Public

Type

Async

See also:

None

int32_t _CWORD87_Lan_LoopbackStop (HANDLE *h_app*, [CWORD87 Lan Cmdhdr](#) * *cmd*)**Brief***CWORD87* -LAN command stop loopback periodical transmission**Parameters:**

in	<i>h_app</i>	HANDLE - Handle for Application
in	<i>cmd</i>	_CWORD87_Lan_Cmd* - Command data

Return values:

<i>LAN_RET_NORMAL</i>	: Success
<i>LAN_RET_ERROR_PARAM</i>	: Paramter error
<i>LAN_RET_ERROR_CANCEL</i>	: Send the message failed

Prerequisite

Availability of Communication is TRUE

Change of internal state

Change of internal state according to the API does not occur.

Classification

Public

Type

Async

See also:

None

int32_t _CWORD87_Lan_StopRegularTransmission (HANDLE *h_app*, [CWORD87 Lan Cmdhdr](#) * *cmd*)**Brief***CWORD87* -LAN command stop periodical transmission**Parameters:**

in	<i>h_app</i>	HANDLE - Handle for Application
in	<i>cmd</i>	_CWORD87_Lan_Cmd* - Command data

Return values:

<i>LAN_RET_NORMAL</i>	: Success
<i>LAN_RET_ERROR_PARAM</i>	: Paramter error
<i>LAN_RET_ERROR_CANCEL</i>	: Send the message failed

Prerequisite

Availability of Communication is TRUE

Change of internal state

Change of internal state according to the API does not occur.

Classification

Public

Type

Async

See also:

None

int32_t _CWORD87_Lan_Transmission (HANDLE *h_app*, uint8_t *com_type*, [CWORD87 Lan Cmd](#) * *cmd*, uint16_t *cycle*)

Brief

CWORD87-LAN command transmission request

Parameters:

in	<i>h_app</i>	HANDLE - Handle for Application
in	<i>com_type</i>	uint8_t - Communication type (LAN_COMTYPE_XXX)
in	<i>cmd</i>	_CWORD87_Lan_Cmd* - Command data
in	<i>cycle</i>	uint16_t - Periodic transmission cycle

Return values:

<i>LAN_RET_NORMAL</i>	: Success
<i>LAN_RET_ERROR_PARAM</i>	: Paramter error
<i>LAN_RET_ERROR_CANCEL</i>	: Send the message failed

Prerequisite

Availability of Communication is TRUE

Change of internal state

Change of internal state according to the API does not occur.

Classification

Public

Type

Async

See also:

None

int32_t _CWORD87_Lan_Transmission_addPhysiAdr (HANDLE *h_app*, uint8_t *com_type*, uint16_t *physical_adr*, [CWORD87 Lan Cmd](#) * *cmd*, uint16_t *cycle*)

Brief

Send CWORD87 -LAN message. (Specify the physical address)

Parameters:

in	<i>h_app</i>	HANDLE - Handle for Application
in	<i>com_type</i>	uint8_t - Communication type (LAN_COMTYPE_XXX)
in	<i>physical_adr</i>	uint16_t - Destination physical address
in	<i>cmd</i>	_CWORD87_Lan_Cmd* - Command data
in	<i>cycle</i>	uint16_t - Periodic transmission cycle

Return values:

<i>LAN_RET_NORMAL</i>	: Success
<i>LAN_RET_ERROR_PARAM</i>	: Paramter error
<i>LAN_RET_ERROR_CANCEL</i>	: Send the message failed

Prerequisite

Availability of Communication is TRUE

Change of internal state

Change of internal state according to the API does not occur.

Classification

Public

Type

Async

See also:

None