AIS HOTEL DLL API

D-LOGIC
Version 4.x
25.03.2014

Table of Contents

File Index

File List

Here is a list of all files with brief descriptions:

File Documentation

ais_hotel/include/ais_hotel.h File Reference

Typedefs

typedef char * **cstr**

type of pointer to string

Enumerations

enum **e_card_action** { **ACTION_CARD_FOREIGN** = 0x00, **ACTION_CARD_DISCARDED** = 0x20, **ACTION_CARD_HACKED** = 0x40, **ACTION_CARD_BAD_DATA** = 0x50, **ACTION_CARD_NO_DATA** = 0x60, **ACTION_CARD_UNLOCKED** = 0x80, **ACTION_CARD_UNLOCKED_1** = 0x81, **ACTION_CARD_UNLOCKED_2** = 0x82, **ACTION_CARD_UNLOCKED_3** = 0x83, **ACTION_CARD_UNLOCKED_4** = 0x84, **ACTION_CARD_UNLOCKED_5** = 0x85, **ACTION_CARD_UNLOCKED_6** = 0x86, **ACTION_CARD_UNLOCKED_7** = 0x87 }

Functions

DL_API **DL_STATUS AIS_Hotel_CommunicationOpen** (void)
DL_API **DL_STATUS AIS_Hotel_CommunicationClose** (void)
DL_API **DL_STATUS AIS_Hotel_GetStatus** (int reader_id, int *is_power_on, int *unreaded_rte, unsigned char status_data[5])
DL_API **DL_STATUS AIS_Hotel_TimeGet** (int reader_id, int *tm_year, int *tm_mon, int *tm_mday, int *tm_hour, int *tm_min, int *tm_sec)
DL_API **DL_STATUS AIS_Hotel_TimeSet** (int reader_id, const char password[], int tm_year, int tm_mon, int tm_mday, int tm_hour, int tm_min, int tm_sec)
DL_API **DL_STATUS AIS_Hotel_CardBlock** (int reader_id, const char password[], int card_sn)
DL_API **DL_STATUS AIS_Hotel_CardUnblock** (int reader_id, const char password[], int card_sn)
DL_API **DL_STATUS AIS_Hotel_CardUnblockAll** (int reader_id, const char password[])
DL_API **DL_STATUS AIS_Hotel_GetLastRTE** (int reader_id, int *new_log, int *action, int *card_id, int *system_id, int *tm_year, int *tm_mon, int *tm_mday, int *tm_hour, int *tm_min, int *tm_sec)
DL_API const **cstr AIS_GetDLLVersion** (void)
DL_API **cstr dbg_status2str** (**DL_STATUS** status)
DL_API **cstr dbg_action2str** (**e_card_action** action)

Typedef Documentation

typedef char* cstr

type of pointer to string

pointer to character array

Definition at line 201 of file ais_hotel.h.

Enumeration Type Documentation

enum e_card_action

Enumerator

ACTION_CARD_FOREIGN
ACTION_CARD_DISCARDED
ACTION_CARD_HACKED
ACTION_CARD_BAD_DATA
ACTION_CARD_NO_DATA
ACTION_CARD_UNLOCKED
ACTION_CARD_UNLOCKED_1
ACTION_CARD_UNLOCKED_2
ACTION_CARD_UNLOCKED_3
ACTION_CARD_UNLOCKED_4
ACTION_CARD_UNLOCKED_5
ACTION_CARD_UNLOCKED_6
ACTION_CARD_UNLOCKED_7

Definition at line 29 of file ais_hotel.h.

Function Documentation

DL_API DL_STATUS AIS_Hotel_CommunicationOpen (void)

Open FTDI USB RS485 communication port. Be careful - connect only one FTDI RS485 unit to computer.

You must open port if you need to communicate with a device.

Returns:

DL_STATUS

See Also:

DL_STATUS enum

DL_API DL_STATUS AIS_Hotel_CommunicationClose (void)

Close and release FTDI USB RS485 communication port (if opened).

Returns:

DL_STATUS

See Also:

DL_STATUS enum

DL_API DL_STATUS AIS_Hotel_GetStatus (int *reader_id*, int * *is_power_on*, int * *unreaded_rte*, unsigned char *status_data*[5])

Check relevant statuses of the desired reader.

Parameters:

in	*reader_id*	The reader ID : from 1 to 65534
out	*is_power_on*	Return TRUE (not null) if the reader detect active electric power
out	*unreaded_rte*	Return how many RTE (Real Time Events) is unread from the reader
out	*status_data*	For debug purpose only - send NULL.

Returns:

DL_STATUS

See Also:

DL_STATUS enum

DL_API DL_STATUS AIS_Hotel_TimeGet (int *reader_id*, int * *tm_year*, int * *tm_mon*, int * *tm_mday*, int * *tm_hour*, int * *tm_min*, int * *tm_sec*)

Get actual date and time from the reader clock.

Parameters:

in	*reader_id*	The reader ID : from 1 to 65534
out	*tm_year*	2000 - 2100
out	*tm_mon*	1 - 12
out	*tm_mday*	1 - 31
out	*tm_hour*	0 - 23
out	*tm_min*	0 - 59
out	*tm_sec*	0 - 59

Returns:

DL_STATUS

See Also:

DL_STATUS enum

DL_API DL_STATUS AIS_Hotel_TimeSet (int *reader_id*, const char *password*[], int *tm_year*, int *tm_mon*, int *tm_mday*, int *tm_hour*, int *tm_min*, int *tm_sec*)

Set date and time at the reader clock.

Parameters:

in	*reader_id*	The reader ID : from 1 to 65534
in	*password*	Actual reader password - send as C-string (null terminated string)
in	*tm_year*	2000 - 2100
in	*tm_mon*	1 - 12
in	*tm_mday*	1 - 31
in	*tm_hour*	0 - 23
in	*tm_min*	0 - 59
in	*tm_sec*	0 - 59

Returns:

DL_STATUS

See Also:

DL_STATUS enum

DL_API DL_STATUS AIS_Hotel_CardBlock (int *reader_id*, const char *password*[], int *card_sn*)

Block mifare card with serial number on the reader with specific ID.

Parameters:

in	*reader_id*	The reader ID : from 1 to 65534
in	*password*	Actual reader password - send as C-string (null terminated string)
in	*card_sn*	Serial number of card which want to block - reader will not pass that card anymore

Returns:

DL_STATUS

See Also:

DL_STATUS enum

DL_API DL_STATUS AIS_Hotel_CardUnblock (int *reader_id*, const char *password*[], int *card_sn*)

Unblock mifare card with serial number on the reader with specific ID.

Parameters:

in	*reader_id*	The reader ID : from 1 to 65534
in	*password*	Actual reader password - send as C-string (null terminated string)
in	*card_sn*	Serial number of card which want to unblock

Returns:

DL_STATUS

See Also:

DL_STATUS enum

DL_API DL_STATUS AIS_Hotel_CardUnblockAll (int *reader_id*, const char *password*[])

Reset whole black list to allow all cards (all serial numbers)

Parameters:

in	*reader_id*	The reader ID : from 1 to 65534
in	*password*	Actual reader password - send as C-string (null terminated string)

Returns:

DL_STATUS

See Also:

DL_STATUS enum

DL_API DL_STATUS AIS_Hotel_GetLastRTE (int *reader_id*, int * *new_log*, int * *action*, int * *card_id*, int * *system_id*, int * *tm_year*, int * *tm_mon*, int * *tm_mday*, int * *tm_hour*, int * *tm_min*, int * *tm_sec*)

Get one RTE (real time event) if exist from specific reader

Parameters:

in	*reader_id*	The reader ID : from 1 to 65534
out	*new_log*	return not zero value if this function really read RTE from the reader - in this case follow parameters are valid
out	*action*	give what action specific card was created on the reader
out	*card_id*	give ID of checked Mifare card
out	*system_id*	give the system ID if the card is from another system (Job ID)
out	*tm_year*	2000 - 2100

out	*tm_mon*	1 - 12
out	*tm_mday*	1 - 31
out	*tm_hour*	0 - 23
out	*tm_min*	0 - 59
out	*tm_sec*	0 - 59

Returns:

DL_API const cstr AIS_GetDLLVersion (void)

Get information about DLL version

Returns:

Pointer to static C-string (null terminated string), eg. AIS HOTEL DLL version= 4.0.0

DL_API cstr dbg_status2str (DL_STATUS *status*)

Function for debug purpose. Convert DL_STATUS value to string.

Parameters:

in	*status*	

Returns:

Pointer to static C-string (null terminated string) with DL_STATUS description

DL_API cstr dbg_action2str (e_card_action *action*)

Function for debug purpose. Convert card action value to string.

Parameters:

action	

Returns:

ais_hotel/include/dl_status.h File Reference

Enumerations

enum **DL_STATUS** { **DL_OK** = 0x00, **TIMEOUT_ERROR**, **CRC_ERROR**, **NULL_POINTER**, **PARAMETERS_ERROR**, **MEMORY_ALLOCATION_ERROR**, **NOT_INITIALIZED**, **ALREADY_INITIALIZED**, **BUFFER_OVERFLOW**, **TIMESTAMP_INVALID**, **EVENT_BUSY**, **RESOURCE_NOT_ACQUIRED**, **RESOURCE_ALREADY_ACQUIRED**, **RESOURCE_BUSY**, **FILE_OVERSIZE**, **FILE_EMPTY**, **FILE_LOCKED**, **DEVICE_OPENING_ERROR**, **DEVICE_CAN_NOT_OPEN**, **DEVICE_ALREADY_OPENED**, **DEVICE_NOT_OPENED**, **DEVICE_INDEX_OUT_OF_BOUND**, **DEVICE_CLOSE_ERROR**, **LOG_BUFFER_OVERFLOW**, **LOG_EMPTY**, **LOG_NOT_CORRECT**, **CMD_RESPONSE_UNKNOWN_COMMAND**, **CMD_RESPONSE_WRONG_CMD**, **CMD_RESPONSE_COMMAND_FAILED**, **CMD_RESPONSE_NO_AUTHORIZATION**, **CMD_RESPONSE_UNSUCCESS**, **CMD_RESPONSE_SIZE_OVERFLOW**, **CMD_BRAKE_RTE**, **ITEM_IS_ALREADY_IN_LIST**, **ITEM_NOT_IN_LIST**, **THREAD_FAILURE**, **ERR_OBJ_NOT_CREATED**, **DL_HAMMING_NOT_ACK**, **DL_HAMMING_WRONG_ACK**, **DL_HAMMING_WRONG_REPLAY**, **DL_HAMMING_TERR_TIMEOUT**, **DL_HAMMING_TERR_BAD_FRAME**, **DL_HAMMING_TERR_BAD_SUM**, **DL_HAMMING_TERR_BAD_CODE**, **DL_HAMMING_TERR_TOO_OLD**, **DL_HAMMING_TERR_NOISE**,

DL_HAMMING_TERR_ERROR_MASK, **ERROR_SOME_REPLAY_FALULT**, **NO_DEVICES**, **NO_FTDI_COMM_DEVICES** = 0x10000, **NO_FTDI_COMM_DEVICES_OPENED**, **ERR_FTDI_READ**, **ERR_FTDI_READ_LESS_DATA**, **ERR_FTDI_WRITE**, **ERR_FTDI_WRITE_LESS_DATA**, **DL_FT_ERROR_SET_TIMEOUT**, **DL_FT_** = 0x10010, **DL_FT_INVALID_HANDLE**, **DL_FT_DEVICE_NOT_FOUND**, **DL_FT_DEVICE_NOT_OPENED**, **DL_FT_IO_ERROR**, **DL_FT_INSUFFICIENT_RESOURCES**, **DL_FT_INVALID_PARAMETER**, **DL_FT_INVALID_BAUD_RATE**, **DL_FT_DEVICE_NOT_OPENED_FOR_ERASE**, **DL_FT_DEVICE_NOT_OPENED_FOR_WRITE**, **DL_FT_FAILED_TO_WRITE_DEVICE**, **DL_FT_EEPROM_READ_FAILED**, **DL_FT_EEPROM_WRITE_FAILED**, **DL_FT_EEPROM_ERASE_FAILED**, **DL_FT_EEPROM_NOT_PRESENT**, **DL_FT_EEPROM_NOT_PROGRAMMED**, **DL_FT_INVALID_ARGS**, **DL_FT_NOT_SUPPORTED**, **DL_FT_OTHER_ERROR**, **DL_FT_DEVICE_LIST_NOT_READY**, **READER_ERRORS_** = 0x1100, **READER_UID_ERROR**, **READER_LOG_ERROR**, **UNKNOWN_ERROR** = 0xFFFFFFFD, **NOT_IMPLEMENTED** = 0xFFFFFFFE, **MAX_DL_STATUS** = 0xFFFFFFFF, **LAST_ERROR** = 0xFFFFFFFF }

Enumeration Type Documentation

enum DL_STATUS

Enumerator

DL_OK
TIMEOUT_ERROR
CRC_ERROR
NULL_POINTER
PARAMETERS_ERROR
MEMORY_ALLOCATION_ERROR
NOT_INITIALIZED
ALREADY_INITIALIZED
BUFFER_OVERFLOW
TIMESTAMP_INVALID
EVENT_BUSY
RESOURCE_NOT_ACQUIRED
RESOURCE_ALREADY_ACQUIRED
RESOURCE_BUSY
FILE_OVERSIZE
FILE_EMPTY
FILE_LOCKED
DEVICE_OPENING_ERROR
DEVICE_CAN_NOT_OPEN
DEVICE_ALREADY_OPENED
DEVICE_NOT_OPENED
DEVICE_INDEX_OUT_OF_BOUND
DEVICE_CLOSE_ERROR
LOG_BUFFER_OVERFLOW
LOG_EMPTY
LOG_NOT_CORRECT
CMD_RESPONSE_UNKNOWN_COMMAND
CMD_RESPONSE_WRONG_CMD
CMD_RESPONSE_COMMAND_FAILED
CMD_RESPONSE_NO_AUTHORIZATION

CMD_RESPONSE_UNSUCCESS

CMD_RESPONSE_SIZE_OVERFLOW

CMD_BRAKE_RTE

ITEM_IS_ALREADY_IN_LIST

ITEM_NOT_IN_LIST

THREAD_FAILURE

ERR_OBJ_NOT_CREATED

DL_HAMMING_NOT_ACK

DL_HAMMING_WRONG_ACK

DL_HAMMING_WRONG_REPLAY

DL_HAMMING_TERR_TIMEOUT

DL_HAMMING_TERR_BAD_FRAME

DL_HAMMING_TERR_BAD_SUM

DL_HAMMING_TERR_BAD_CODE

DL_HAMMING_TERR_TOO_OLD

DL_HAMMING_TERR_NOISE

DL_HAMMING_TERR_ERROR_MASK

ERROR_SOME_REPLAY_FALULT

NO_DEVICES

NO_FTDI_COMM_DEVICES

NO_FTDI_COMM_DEVICES_OPENED

ERR_FTDI_READ

ERR_FTDI_READ_LESS_DATA

ERR_FTDI_WRITE

ERR_FTDI_WRITE_LESS_DATA

DL_FT_ERROR_SET_TIMEOUT

DL_FT_

DL_FT_INVALID_HANDLE

DL_FT_DEVICE_NOT_FOUND

DL_FT_DEVICE_NOT_OPENED

DL_FT_IO_ERROR

DL_FT_INSUFFICIENT_RESOURCES

DL_FT_INVALID_PARAMETER

DL_FT_INVALID_BAUD_RATE

DL_FT_DEVICE_NOT_OPENED_FOR_ERASE

DL_FT_DEVICE_NOT_OPENED_FOR_WRITE

DL_FT_FAILED_TO_WRITE_DEVICE

DL_FT_EEPROM_READ_FAILED

DL_FT_EEPROM_WRITE_FAILED

DL_FT_EEPROM_ERASE_FAILED

DL_FT_EEPROM_NOT_PRESENT

DL_FT_EEPROM_NOT_PROGRAMMED

DL_FT_INVALID_ARGS

DL_FT_NOT_SUPPORTED

DL_FT_OTHER_ERROR

DL_FT_DEVICE_LIST_NOT_READY

READER_ERRORS_

READER_UID_ERROR

READER_LOG_ERROR

UNKNOWN_ERROR

NOT_IMPLEMENTED

MAX_DL_STATUS

LAST_ERROR

Definition at line 14 of file dl_status.h.

Index

INDEX