CloudGate
libSMS reference manual

Publication date: February 11, 2019
Revision: V013ext

Contents

Chapter 1

Todo List

Class sms_pdu_data_t

move decoded text to the back

Global sms_pdu_parse (char pdu_hex)

change the return value into a pointer that needs to be freed by the user instead of putting it on the stack

change pdu_hex into a unsigned char buffer and an extra size param (int)

Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 5

Module Documentation

5.1 SMS_CDMA

Data Structures

- struct sms_cdma_data_t
- struct sms_pdu_data_t

  structure returned by parsing, containing pdu information

Enumerations

- enum sms_cdma_coding_t {
  SMS_CDMA_CODING_OCTET = 0, SMS_CDMA_CODING_EXTENDED_PROTOCOL_MSG = 1, SMS_CDMA_CODING_7BIT_ASCII = 2, SMS_CDMA_CODING_IA5 = 3,
  SMS_CDMA_CODING_UNICODE = 4, SMS_CDMA_CODING_SHIFT_JIS = 5, SMS_CDMA_CODING_KOREAN = 6, SMS_CDMA_CODING_LATIN_HEBREW = 7,
  SMS_CDMA_CODING_LATIN = 8, SMS_CDMA_CODING_GSM_7BIT = 9, SMS_CDMA_CODING_GSM_DATA_CODING = 10, SMS_CDMA_CODING_UNKNOWN = 0xFF }

  encoding seen by the parser

Functions

- uint8_t sms_cdma_create (char msg, char number, uint16_t id, int flash, int force_ucs2, uint8_t buf)
- sms_cdma_data_t cdma_pdu_parse (uint8_t pdu, uint32_t size)

5.1.1 Detailed Description

5.1.2 Enumeration Type Documentation

5.1.2.1 enum sms_cdma_coding_t

encoding seen by the parser

Todo support other encoding then 7bit ascii

Enumerator

SMS_CDMA_CODING_OCTET (NOT DECODED)

SMS_CDMA_CODING_EXTENDED_PROTOCOL_MSG (NOT DECODED)
SMS_CDMA_CODING_7BIT_ASCII 7bit ASCII
SMS_CDMA_CODING_IA5 (NOT DECODED)
SMS_CDMA_CODING_UNICODE (NOT DECODED)
SMS_CDMA_CODING_SHIFT_JIS (NOT DECODED)
SMS_CDMA_CODING_KOREAN (NOT DECODED)
SMS_CDMA_CODING_LATIN_HEBREW (NOT DECODED)
SMS_CDMA_CODING_LATIN (NOT DECODED)
SMS_CDMA_CODING_GSM_7BIT (NOT DECODED)
SMS_CDMA_CODING_GSM_DATA_CODING (NOT DECODED)
SMS_CDMA_CODING_UNKNOWN (NOT DECODED)

5.1.3 Function Documentation

5.1.3.1 uint8_t sms_cdma_create (char *msg*, char *number*, uint16_t *id*, int *flash*, int *force_ucs2*, uint8_t *buf*)

Create a (set of) pdu(s)

Parameters

in	*msg*	The message to be encoded
in	*number*	The destination address
in	*id*	unique message id, on first use, the user should select an id using a procedure that minimizes the likehood of reusing the same id (see C.S0015 4.3.1.5)
in	*flash*	(NOT IMPLEMENTED) if 0, a CLASS 1 messaage will be created, if 1, a CLASS 0 message will be created
in	*force_ucs2*	(NOT IMPLEMENTED) if 1, the default encoding will be UCS2

Returns

NULL in case of error, and a pdu_list_t object otherwise, user must free the object after use (and also the data pointers of the pdus)

5.1.3.2 sms_cdma_data_t cdma_pdu_parse (uint8_t *pdu*, uint32_t *size*)

Parse a pdu

Parameters

in	*pdu*	the raw pdu
in	*size*	the size of the raw pdu in octets

Returns

sms_cdma_data_t all information of the supplied sms, needs to be freed by the user after use

5.2 SMS_PDU

Data Structures

- struct sms_data_t
- struct pdu_data_t

  structure containing the actual pdu data
- struct pdu_list_t

  list of pdus returened by pdu encoder
- struct sms_pdu_data_t

  structure returned by parsing, containing pdu information

Enumerations

- enum dcs_coding_t { DCS_CLASS0_DEFAULT_ALPHABET, DCS_CLASS1_DEFAULT_ALPHABET, DCS_CLASS2_DEFAULT_ALPHABET, DCS_CLASS0_UNSUPPORTED_ALPHABET, DCS_CLASS1_UNSUPPORTED_ALPHABET, DCS_CLASS2_UNSUPPORTED_ALPHABET, DCS_UNSUPPORTED_CLASS }

  dcs coding type seen by the parser

Functions

- pdu_list_t sms_pdu_create (char msg, char number, char smsc, uint8_t id, int flash, int force_ucs2)
- sms_data_t sms_pdu_parse (char pdu_hex)
- int addr2scabytes (char smsc_number, unsigned char scaBytes)
- void parse_hex (char hex, unsigned char bin)

5.2.1 Detailed Description

5.2.2 Enumeration Type Documentation

5.2.2.1 enum dcs_coding_t

dcs coding type seen by the parser

Enumerator

DCS_CLASS0_DEFAULT_ALPHABET class 0, default gsm 7bit

DCS_CLASS1_DEFAULT_ALPHABET class 1, default gsm 7bit

DCS_CLASS2_DEFAULT_ALPHABET class 2, default gsm 7bit

DCS_CLASS0_UNSUPPORTED_ALPHABET class 0, unidentified alphabet, could be ucs2

DCS_CLASS1_UNSUPPORTED_ALPHABET class 1, unidentified alphabet, could be ucs2

DCS_CLASS2_UNSUPPORTED_ALPHABET class 2, unidentified alphabet, could be ucs2

DCS_UNSUPPORTED_CLASS an unsupported class was detected (e.g. compressed, message waiting indication, class > 2,...)

5.2.3 Function Documentation

5.2.3.1 pdu_list_t sms_pdu_create (char *msg*, char *number*, char *smsc*, uint8_t *id*, int *flash*, int *force_ucs2*)

Create a (set of) pdu(s)

Parameters

in	*msg*	The message to be encoded
in	*number*	The destination address
in	*smsc*	Optional Service center address
in	*id*	unique message id (used in case of concatenation)
in	*flash*	if 0, a CLASS 1 message will be created, if 1, a CLASS 0 message will be created
in	*force_ucs2*	if 1, the default encoding will be UCS2

Returns

NULL in case of error, and a pdu_list_t object otherwise, user must free the object after use (and also the data pointers of the pdus)

5.2.3.2 sms_data_t sms_pdu_parse (char *pdu_hex*)

Parse a pdu

Parameters

in	*pdu_hex*	the pdu in "hex string format"

Returns

sms_data_t all information of the supplied PDU

Todo change the return value into a pointer that needs to be freed by the user instead of putting it on the stack

change pdu_hex into a unsigned char buffer and an extra size param (int)

5.2.3.3 int addr2scabytes (char *smsc_number*, unsigned char *scaBytes*)

encode a smsc_number into 23.040 scaBytes

Parameters

in	*smsc_number*	a null terminated string containing the smsc
in,out	*scaBytes*	a buffer where the encoded service center will be put

Returns

number of bytes put in scaBytes

5.2.3.4 void parse_hex (char *hex*, unsigned char *bin*)

converts a hex string to binary array

Parameters

in	*hex*	a null terminated string containing hex values
out	*bin*	an array of bytes to store the converted binary values

Chapter 6

Data Structure Documentation

6.1 pdu_data_t Struct Reference

structure containing the actual pdu data

#include <sms_pdu.h>

Data Fields

- int size
- unsigned char data

6.1.1 Detailed Description

structure containing the actual pdu data

6.1.2 Field Documentation

6.1.2.1 int pdu_data_t::size

Size of data in octects

6.1.2.2 unsigned char pdu_data_t::data

23.040 encoded pdu, user must free the data pointer after use

The documentation for this struct was generated from the following file:

- libsms/sms_pdu.h

6.2 pdu_list_t Struct Reference

list of pdus returened by pdu encoder

#include <sms_pdu.h>

Data Fields

- int length
- pdu_data_t pdu []

6.2.1 Detailed Description

list of pdus returened by pdu encoder

6.2.2 Field Documentation

6.2.2.1 int pdu_list_t::length

number of pdus created

6.2.2.2 pdu_data_t pdu_list_t::pdu[]

array of pdus

The documentation for this struct was generated from the following file:

- libsms/sms_pdu.h

6.3 sms_cdma_data_t Struct Reference

Data Fields

- char sender [64]
- unsigned char date [10]
- unsigned char time [10]
- time_t ctime
- unsigned int part_count
- unsigned int part_index
- unsigned int part_refno
- sms_cdma_coding_t coding
- char text [256]

6.3.1 Field Documentation

6.3.1.1 char sms_cdma_data_t::sender[64]

Sender's Address

6.3.1.2 unsigned char sms_cdma_data_t::date[10]

Date of sms

6.3.1.3 unsigned char sms_cdma_data_t::time[10]

Time of sms

6.3.1.4 time_t sms_cdma_data_t::ctime

time_t object of date/time of sms

6.3.1.5 unsigned int sms_cdma_data_t::part_count

(NOT IMPLEMENTED) Sms concatenation: part count (parsed from 23.040 IEI 0x00: 8-bit reference number)

6.3.1.6 unsigned int sms_cdma_data_t::part_index

(NOT IMPLEMENTED) Sms concatenation: part index (parsed from 23.040 IEI 0x00: 8-bit reference number)

6.3.1.7 unsigned int sms_cdma_data_t::part_refno

(NOT IMPLEMENTED) Sms long reference number: part count (parsed from 23.040 IEI 0x00: 8-bit reference number)

6.3.1.8 sms_cdma_coding_t sms_cdma_data_t::coding

Parsed coding type

6.3.1.9 char sms_cdma_data_t::text[256]

Decoded text

The documentation for this struct was generated from the following file:

- libsms/sms_cdma.h

6.4 sms_data_t Struct Reference

Data Fields

- unsigned char serviceCenter [64]
- unsigned char sender [64]
- unsigned char date [10]
- unsigned char time [10]
- unsigned char text [160 4+1]
- time_t ctime
- unsigned int part_count
- unsigned int part_index
- unsigned int part_refno
- unsigned char mti
- unsigned char tp_pid
- dcs_coding_t coding

6.4.1 Field Documentation

6.4.1.1 unsigned char sms_data_t::serviceCenter[64]

Service Center Address if included

6.4.1.2 unsigned char sms_data_t::sender[64]

Sender's Address

6.4.1.3 unsigned char sms_data_t::date[10]

Date of sms

6.4.1.4 unsigned char sms_data_t::time[10]

Time of sms

6.4.1.5 unsigned char sms_data_t::text[160 4+1]

Decoded text

6.4.1.6 time_t sms_data_t::ctime

time_t object of date/time of sms

6.4.1.7 unsigned int sms_data_t::part_count

Sms concatenation: part count (parsed from 23.040 IEI 0x00: 8-bit reference number)

6.4.1.8 unsigned int sms_data_t::part_index

Sms concatenation: part index (parsed from 23.040 IEI 0x00: 8-bit reference number)

6.4.1.9 unsigned int sms_data_t::part_refno

Sms long reference number: part count (parsed from 23.040 IEI 0x00: 8-bit reference number)

6.4.1.10 unsigned char sms_data_t::mti

Message Type indication (23.040 TP-MTI)

6.4.1.11 unsigned char sms_data_t::tp_pid

Protocol Identifier (23.040 TP-PID)

6.4.1.12 dcs_coding_t sms_data_t::coding

Parsed coding type

The documentation for this struct was generated from the following file:

- libsms/sms_pdu.h

6.5 sms_pdu_data_t Struct Reference

structure returned by parsing, containing pdu information

#include <sms_cdma.h>

6.5.1 Detailed Description

structure returned by parsing, containing pdu information

Todo move decoded text to the back

The documentation for this struct was generated from the following file:

- libsms/sms_cdma.h

Chapter 7

File Documentation

7.1 libsms/sms_cdma.h File Reference

Header file with definitions for creating and parsing 3GPP2 (CDMA) Layer 3 SMS (C.S0015)

#include <stdint.h>
#include <time.h>

Data Structures

- struct sms_cdma_data_t

Enumerations

- enum sms_cdma_coding_t { SMS_CDMA_CODING_OCTET = 0, SMS_CDMA_CODING_EXTENDED_PROTOCOL_MSG = 1, SMS_CDMA_CODING_7BIT_ASCII = 2, SMS_CDMA_CODING_IA5 = 3, SMS_CDMA_CODING_UNICODE = 4, SMS_CDMA_CODING_SHIFT_JIS = 5, SMS_CDMA_CODING_KOREAN = 6, SMS_CDMA_CODING_LATIN_HEBREW = 7, SMS_CDMA_CODING_LATIN = 8, SMS_CDMA_CODING_GSM_7BIT = 9, SMS_CDMA_CODING_GSM_DATA_CODING = 10, SMS_CDMA_CODING_UNKNOWN = 0xFF }
  encoding seen by the parser

Functions

- uint8_t sms_cdma_create (char msg, char number, uint16_t id, int flash, int force_ucs2, uint8_t buf)
- sms_cdma_data_t cdma_pdu_parse (uint8_t pdu, uint32_t size)

7.1.1 Detailed Description

Header file with definitions for creating and parsing 3GPP2 (CDMA) Layer 3 SMS (C.S0015)

7.2 libsms/sms_pdu.h File Reference

Header file with definitions for creating and parsing 3GPP (GSM/UMTS) 23.040 PDUs.

#include <stdint.h>
#include <time.h>

Data Structures

- struct sms_data_t
- struct pdu_data_t

  structure containing the actual pdu data
- struct pdu_list_t

  list of pdus returened by pdu encoder

Enumerations

- enum dcs_coding_t { DCS_CLASS0_DEFAULT_ALPHABET, DCS_CLASS1_DEFAULT_ALPHABET, DCS_CLASS2_DEFAULT_ALPHABET, DCS_CLASS0_UNSUPPORTED_ALPHABET, DCS_CLASS1_UNSUPPORTED_ALPHABET, DCS_CLASS2_UNSUPPORTED_ALPHABET, DCS_UNSUPPORTED_CLASS }

  dcs coding type seen by the parser

Functions

- pdu_list_t sms_pdu_create (char msg, char number, char smsc, uint8_t id, int flash, int force_ucs2)
- sms_data_t sms_pdu_parse (char pdu_hex)
- int addr2scabytes (char smsc_number, unsigned char scaBytes)
- void parse_hex (char hex, unsigned char bin)

7.2.1 Detailed Description

Header file with definitions for creating and parsing 3GPP (GSM/UMTS) 23.040 PDUs.

Index