Service Access Point

SIM Driver

Department:	Aalborg Wireless Center
Creation Date:	20 January, 2003
Last Modified:	10 June, 2004 by Flemming Dunker
ID and Version:	8010.136.03.009
Status:	Being Processed

0 Document Control

0.1 Document History

ID	Author	Date	Status
8010.136.03.001	KKS	20 January, 2003	Accepted
8010.136.03.002	KKS	20 January, 2003	Accepted
8010.136.03.003	KKS	20 January, 2003	Accepted
8010.136.03.004	KKS	26 May, 2003	Accepted
8010.136.03.005	KKS	26 May, 2003	Accepted
8010.136.03.006	KKS	20 August, 2003	Accepted
8010.136.03.007	FK	20 November, 2003	Being processed
8010.136.03.008	FDU	3 May, 2004	Being Processed
8010.136.03.009	FDU	10 June, 2004	Being Processed

0.2 References, Abbreviations, Terms

[TI 8010.801] 8010.801, References and Vocabulary, Texas Instruments

Table of Contents

1 Introduction

The interface is function based only.

Throughout the document the term UICC is used to cover both a GSM SIM that contains one single application, the GSM application, and a multi-application UICC that contains one or more applications, maximum one GSM and one or more UMTS applications.

Data stores are used to transfer information between the SIM reader driver and the PS. Pointers to the data stores where the data is written to and read from are handed to the SIM reader driver by the PS. The format of the data stored in these stores are compliant to the format specified in [3G 11.11], [3G 11.14], [3G 31.102], [3G 31.111] and [ETSI 102 221]. Therefore, the formatting of data located in these stores are not described throughout this document. The same approach is used in regards to status words, etc. Only in the case where SIM data is reformatted, will they be described in this document.

Two functions are not defined in this document and are the two only functions called from the SIM driver to the PS. These are the SIM insertion and SIM removal functions. The first call after power-on, the simdrv_register function, gives two function pointers to the SIM driver for these two functions. Else, all function calls are initiated by the PS.

The transmission protocol is handled autonomously by the SIM reader driver. This means that the interface between the PS and SIM reader driver operates at APDU command level and that the PS has no knowledge of which protocol has been selected for communication between the SIM reader driver and the inserted UICC during the PPS procedure. Thus the PS is not required to perform the GET RESPONSE command, instead this is done, in case of protocol T=0, by the driver .

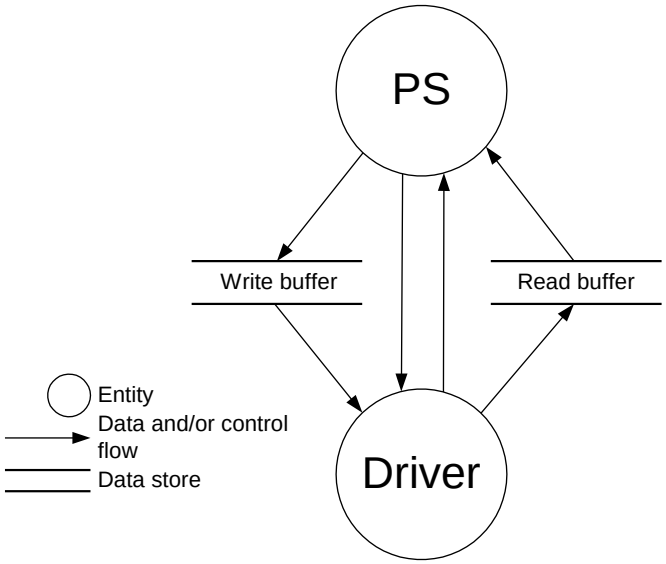

Figure 1 - Interface overview

Communication between the PS and the driver will always be initiated by the PS, with two exceptions to that rule.

- When the PS asks the driver to reset the UICC, the driver uses a call back function that is handed during initialisation to inform the PS.
- When the inserted UICC is physically removed from the PS, the driver calls another call back function that is handed during initialisation

2 Constants

Description:

Defines maximum value used in parameter definitions

Pragma:

Name	Value	Comment
PREFIX	SIMDRV	Prefix for this document
ALLWAYS_ENUM_IN_VAL_FILE	YES	Adds enumerations in the .val file.
ENABLE_GROUP	YES	Enable h-file grouping
COMPATIBILITY_DEFINES	NO	Compatible to the old #defines

Definition:

Name	Value	Comment	Group
SIZE_ATR_INFO	0x21	The maximum length of the data returned from the ATR procedure	uicc
MIN_RESULT	0x01	Minimum size of the response of any given command	uicc
MAX_RESULT	0x100	Maximum size of the response of any given command	uicc
MIN_DATA_SIZE	0x01	Minimum length of a data element	uicc
MAX_DATA_SIZE	0xFF	Maximum length of a data element	uicc
MAX_READERS	0x02	Maximum number of card readers supported.	uicc

History:

20-February-02	FDU	Initial
30-October-2003	FK	'MAX_RESULT' corrected
03-May-04	FDU	PREFIX changed to SIMDRV

3 Primitives

3.1 SIMDRV_DUMMY_PRIM

Description:

Dummy.

Definition:

Short Name	ID	Direction	Group
SIMDRV_DUMMY	0x80FF009A	SIM -> DRIVER	uicc

Elements:

Long Name	Short Name	CTRL	Ref	Type
Insert call-back function pointer	insert	PTR	5.1	STRUCT
Remove call-back function pointer	remove	PTR	5.2	STRUCT
SIM Card Info	atr_string_info	PTR	5.3	STRUCT
Return Value	reset_return_val		5.4	U8
Result Buffer	result_info	PTR	5.5	STRUCT
Length of expected data	len		5.6	U16
Update Data Element	data_info	PTR	5.7	STRUCT
Status Words	sw1_2		5.8	U16
Class Byte	cla		5.9	U8
Instruction Code	ins		5.10	U8
Transparent command header	cmd_header		5.11	STRUCT
Perform Voltage Selection	voltage_select		5.12	U8
SIM Command Parameters	p1		5.13	U8
SIM Command Parameters	p2		5.13	U8
Configuration Characteristics requested	config_requested		5.14	U8
Configuration Characteristics	config_characteristics	PTR	5.15	STRUCT
Reader Id	reader_id		5.16	U8
UICC Characteristics	uicc_characteristics		5.17	U8

History:

9-December-2002	BRY	Initial version after merge.
20-November-2003	FK	New prefix; changes in section 5.
8-Dec-2003	FK	Revised
03-May-04	FDU	Aligned with contents of chapter 4 and chapter 5
10-June-04	FDU	Aligned with contents of chapter 4 and chapter 5

4 Functions

4.1 simdrv_register

Description:

The function is used to establish communication between the PS and the SIM reader driver. Two call back function pointers are passed to the driver. There is no return value.

The call back to 'insert()' is executed by the driver, when it, being called by the function 'simdrv_reset()', is able to successfully reset the SIM card.

The call back 'remove()' is executed by the driver at any time during a card session, when it detects that the SIM card cannot be accessed any more.

Definition:

Short Name	ID	Direction	Group
void simdrv_register (void (* insert) (T_SIMDRV_atr_string_info *atr_string_info, U8 config_requested , T_SIMDRV_config_characteristics * config_characteristics), void (* remove)(void))	InlineC	SIM->DRIVER	none

Elements:

Long Name	Short Name	CTRL	Ref	Type
Insert call-back function pointer	insert	PTR	5.1	STRUCT
Remove call-back function pointer	remove	PTR	5.2	STRUCT
Answer To Reset string	atr_string_info	PTR	5.3	STRUCT
Configuration Characteristics requested	config_requested		5.14	U8
Configuration Characteristics	config_characteristics	PTR	5.15	STRUCT

History:

20-February-2002	FDU	Initial.
20-November-2003	FK	Parameters for callbacks added.
03-May-04	FDU	function renamed, parameters aligned to review report
10-June-04	FDU	config_request parameters added.
19-Aug-04	FDU	(void) added to remove pointer.

4.2 simdrv_poweroff

Description:

This function is used to inform the SIM reader driver that it shall deactivate. Nothing is passed to the SIM hardware driver, neither is something returned.

Definition:

Short Name	ID	Direction	Group
void simdrv_poweroff (U8 reader_id)	InlineC	SIM->DRIVER	none

Elements:

Long Name	Short Name	CTRL	Ref	Type
Reader Id	reader_id		5.16	U8

History:

20-February-2002	FDU	Initial.
20-November-2003	FK	Prefix changed; description extended.
03-May-04	FDU	function renamed
10-June-04	FDU	reader id added

4.3 simdrv_reset

Description:

The PS uses this function to make the driver attempt a reset of the inserted UICC and to start card session.

If the driver does not detect the presence of a SIM card, then it shall return the value SIM_NOT_INSERTED.

If the driver is able to successfully initialise the card and start a card session, it calls the call-back function 'insert()' with the appropriate parameters. After regaining control from the call-back function the driver finishes the reset function and returns the value SIM_INSERTED.

The function has a parameter which is used to indicate to the SIM reader driver whether or not the SIM reader driver shall reset the card performing the voltage selection procedure or reset the card using the current voltage class.

Definition:

Short Name	ID	Direction	Group
U8 simdrv_reset (U8 reader_id , U8 voltage_select)	InlineC	SIM->DRIVER	none

Elements:

Long Name	Short Name	CTRL	Ref	Type
Reader Id	reader_id		5.16	U8
Perform voltage selection	voltage_select		5.12	U8
Return Value	reset_return_val		5.4	U8

History:

20-February-2002	FDU	Initial.
16-December-2002	FDU	Parameters changed to pure struct pointers.
03-May-04	FDU	function renamed, parameters aligned to review report
10-June-04	FDU	reader id added

4.4 simdrv_xch_apdu

Description:

Used to transparently exchange APDU commands and result with any application, which may reside on the SIM. If the communication between SIM and ME is run by the protocol T=0, then the functions performs autonomously the GET RESPONSE Command observing the following rule:

- If the card responds with SW1=9F, then the command shall be coded as described in [GSM 11.11].
- If the card responds with SW1=61, then the command shall be coded as described in [ETSI TS 102 221].

The function complies with the following rules for the interpretation of its parameters:

- Access to a NULL pointer, either read or write, must not be performed. If this leads to a

contradiction (i.e. a data count is given, but the null pointer prevents access to the data), then the function shall return SIMDRV_ERR_PARAM_WRONG.

- If parameter 'result_info' is set to ZERO or its element 'len' is set to ZERO, then the driver shall not read data from the card. If the card answers with SW1=61/9F, then this result code is passed to the caller, but GET RESPONSE is not performed.
- If the element 'len' of parameter 'result_info' is set to NON-ZERO, then the driver shall try to read up to 'len' bytes from the card. The data is stored in element 'result', the count element receives the number of actually read bytes. In case of protocol T=0 the element 'len' is identical with parameter P3, in case of protocol T=1 it is identical with parameter Le. If the UICC responds with SW1=6C and a requested size larger than indicated in ‘len’, it shall still only return the amount of data indicated in ‘len’, discarding the remain.
- If the count element of parameter 'data_info' is set to NON-ZERO and its element 'data' is not set to ZERO, then the driver shall send the data stored in element 'data' up to the number given by the count element to the card. In case of protocol T=0 the count element is identical with parameter P3, in case of protocol T=1 it is identical with parameter Lc.
- If neither send nor receive data is given by the function parameters, then,. in case of protocol T=0, the function shall only send the command header to the card followed by parameter P3 set to ZERO. In case of protocol T=1 only the header is sent.

Definition:

Short Name	ID	Direction	Group
U16 simdrv_xch_apdu (U8 reader_id , T_SIMDRV_cmd_header cmd_header, T_SIMDRV_data_info data_info, T_SIMDRV_result_info * result_info)	InlineC	SIM->DRIVER	none

Elements:

Long Name	Short Name	CTRL	Ref	Type
Reader Id	reader_id		5.16	U8
Transparent command header	cmd_header		5.11	STRUCT
Send Data Buffer	data_info	PTR	5.7	STRUCT
Result Buffer	result_info	PTR	5.5	STRUCT
Status Words	sw1_2		5.8	U16

History:

30-October-2003	FK	Initial.
03-May-04	FDU	function renamed, parameters aligned to review report
10-June-04	FDU	reader id added

5 Parameters

5.1 Insert call-back function pointer

Description:

The insert call back function to which the pointer refers shall be executed by the driver after the PS has executed the simdrv_reset() command call, if and only if the activation of the inserted UICC was successful. It has a two parameters. One used to indicate to the PS whether UICC configuration characteristics are required by the driver and one which points to a structure, where the PS can store the retrieved configuration characteristics. (See chapter 6)

Definition:

Type	Short Name	Comment	Group
STRUCT	insert	function pointer	uicc

Elements:

Long Name	Short Name	Ref	Type
insert pointer	insert_ptr		U32

History:

21-February-02	FDU	Initial
03-May-04	FDU	function renamed, parameters aligned to review report

5.2 Remove call-back function pointer

Description:

The remove call back function to which the pointer refers, shall be executed by the driver when the driver detects that the inserted UICC has been removed. Either this detection is due to missing reply to command or due to hardware removal detection.

Definition:

Type	Short Name	Comment	Group
STRUCT	remove	function pointer	uicc

Elements:

Long Name	Short Name	Ref	Type
remove pointer	remove_ptr		U32

History:

21Feb-02	FDU	Initial
03-May-04	FDU	function renamed, parameters aligned to review report

5.3 SIM Card Information

Description:

Contains the result of the ATR procedure, which the SIM hardware driver initiates during its start up sequence.

Definition:

Type	Short Name	CTRL	Comment	Group

STRUCT	atr_string_info	PTR	SIM Card Info	uicc

Elements:

Long Name	Short Name	CTRL	Ref	Type
SIM Card Answer to reset string	atr_string	[2..SIZE_ATR_INFO]		U8

History:

20-February-02	FDU	Initial
16-December-02	FDU	Structure changed for element to a struct including a length byte
03-May-04	FDU	function renamed, parameters aligned to review report

5.4 Reset return value

Description:

The return value is used by the SIM hardware driver to indicate to the PS whether a card is inserted or not and if it is fully functioning.

Definition:

Type	Short Name	Comment	Group
U8	reset_return_val	Reset return value	uicc

Values:

Value	C-Macro	Comment
0	SIM_INSERTED	A SIM is inserted and ATR/PPS was successful
1	SIM_NOT_INSERTED	No SIM inserted (hardware detected)
2	INVALID_CARD	Card is not responding or gives unintelligible answers, communications time out.
3	ME_FAILURE	Power management related problems, e.g. voltage selection
4	IMPROPER_CALL_BACK	simdrv_register function called with pointer improperly set, e.g. NULL.
5	ME_READER_NOT_AVAILABLE	The reader requested is not available

History:

20-February-02	FDU	Initial
05-July-02	FDU	Meaning of byte changed
03-May-04	FDU	function renamed, parameters aligned to review report
28-June-04	FDU	Added latter two return values after desk check.

5.5 Result Data

Description:

The SIM driver uses this to return the response data retrieved when issuing a command.

Definition:

Type	Short Name	CTRL	Comment	Group
STRUCT	result_info	PTR	Result Buffer	uicc

Elements:

Long Name	Short Name	CTRL	Ref	Type

Length of expected data	len		5.6	U16
Result byte	result	DYN[MIN_RESULT..MAX_RESULT]		U8

History:

20-February-02	FDU	Initial
16-December-02	FDU	Structure changed for element to a struct including a length byte
03-May-04	FDU	function renamed, parameters aligned to review report

5.6 Length of Expected Data

Description:

Used to indicate maximum length, in bytes, of data which the ME expects the UICC to return, e.g. during a READ_RECORD or a READ_BINARY command. In cases where the size of the return data is undeterminable, the value shall be LENGTH_UNKNOWN (See chapter 6)

Definition:

Type	Short Name	Comment	Group
U16	len	Maximum length of expected data	uicc

Values:

Value	C-Macro	Comment
0x00-0x100		Range of Length of Data
0xFFFF	LENGTH_UNKNOWN	Indicating that the length expected is unknown.

History:

21-February-02	FDU	Initial
03-May-04	FDU	function renamed, parameters aligned to review report

5.7 Send Data

Description:

The Update data element is used by the PS to transfer the data which it wants to update to the SIM hardware driver.

Definition:

Type	Short Name	CTRL	Comment	Group
STRUCT	data_info	PTR	Data element info	uicc

Elements:

Long Name	Short Name	CTRL	Ref	Type
Data element	data	DYN[MIN_DATA_SIZE..MAX_DATA_SIZE]		U8

History:

20-February-02	FDU	Initial
16-December-02	FDU	Structure changed for element to a struct including a length byte
03-May-04	FDU	function renamed, parameters aligned to review report

5.8 Status Words

Description:

After each command issued by the ME to the UICC, the SIM hardware driver shall return the two status bytes from the UICC. The PS interprets these value and acts upon them on its own. The possible values of the status word can be found in [GSM 11.11] and [ETSI TS 102 221]. They are returned in a U16, SW1 occupies bits 8..15 and SW2 occupies bits 0..7 of the value.
Additionally there are a couple of driver error codes which are returned in case of unrecoverable communication problems between ME and UICC. Before the driver returns any of these error codes, it shall retry the causing operation at least once.

Definition:

Type	Short Name	Comment	Group
U16	sw1_2	Status Words	uicc

Values:

Value	C-Macro	Comment
0x0001	ERR_NOCARD	No SIM inserted (hardware detected)
0x0002	ERR_NOT_RESET	A reset has not been performed on the driver with the requested Id.
0x0003	ERR_ME_FAIL	Unrecoverable ME failure (for instance interrupt fails to occur)
0x0004	ERR_RETRY_FAIL	ME/SIM communication failed after certain retries, SIM reset required
0x0005	ERR_PARAM_WRONG	A driver function is called with invalid parameters

History:

21-February-02	FDU	Initial
8-Dec-2003	FK	Revised
03-May-04	FDU	function renamed, parameters aligned to review report

5.9 Class byte

Description:

When communicating with an UMTS Application, the Class Byte shall be set according to [ETSI 102 221, 10.1.2], either code 0x00 (standard ISO 7816-4 Class Byte) or code 0x80 (for additional instructions defined in ETSI 102 221) dependent on the type of the selected command. For a GSM application according to GSM 11.11 / 3GPP 51.011 the code 0xA0 is used for all instructions.

Definition:

Type	Short Name	Comment	Group
U8	cla	Class Byte	uicc

Values:

Value	C-Macro	Comment
0xA0	GSM_CLASS_BYTE	GSM Class byte according to [GSM 11.11 / 3GPP 51.011]
0x80	UMTS_CLASS_BYTE	UMTS Class byte according to [ETSI 102 221 10.1.2]
0x00	UICC_CLASS_BYTE	UICC Class byte according to [ISO 7816-4]

History:

4-July-2002	FDU	Initial
5-Dec-2003	FK	Revised

03-May-04	FDU	function renamed, parameters aligned to review report

5.10 SIM Instruction Code

Description:

The parameter defines the SIM instruction code for SIM access.

Definition:

Type	Short Name	Comment	Group
U8	ins	instruction code	uicc

Values:

Value	C-Macro	Comment
0xA4	INS_SELECT	Select Instruction
0xF2	INS_STATUS	Status Instruction
0xB0	INS_READ_BINARY	Read Binary Instruction
0xD6	INS_UPDATE_BINARY	Update Binary Instruction
0xB2	INS_READ_RECORD	Read Record Instruction
0xDC	INS_UPDATE_RECORD	Update Record Instruction
0xA2	INS_SEEK	Seek Instruction
0x32	INS_INCREASE	Increase Instruction
0x20	INS_VERIFY_CHV	Verify Chv Instruction
0x24	INS_CHANGE_CHV	Change Chv Instruction
0x26	INS_DISABLE_CHV	Disable Chv Instruction
0x28	INS_ENABLE_CHV	Ensable Chv Instruction
0x2C	INS_UNBLOCK_CHV	Unblock Chv Instruction
0x04	INS_INVALIDATE	Invalidate Instruction
0x44	INS_REHABILITATE	Rehabilitate Instruction
0x88	INS_AUTHENTICATE	Run GSM Algorithm (2/2.5G) / Authenticate (3G) Instruction
0x10	INS_TERMINAL_PROFILE	Terminal Profile Instruction
0xC2	INS_ENVELOPE	Envelope Instruction
0x12	INS_FETCH	Fetch Instruction
0x14	INS_TERMINAL_RESPONSE	Terminal Response Instruction
0xC0	INS_GET_RESPONSE	Get Response Instruction

History:

24-April-2002	JK	Initial
01-May-03	KBS	SIM_ prefix removed due to autoprefixing
8-Dec-2003	FK	Incorporated from SAP SIM

5.11 Command Header

Description:

A struct pointer to create a command header for sending transparent commands to the UICC

Definition:

Type	Short Name	Comment	Group
STRUCT	cmd_header	Transparent command header	uicc

Elements:

Long Name	Short Name	CTRL	Ref	Type
Class Byte value	cla		5.9	U8
Instruction Code	ins		5.10	U8
Parameter P1	p1		5.13	U8

Parameter P2	p2	5.13	U8

History:

30-October-03 FK Initial

5.12 Perform voltage selection

Description:

Used to indicate to the SIM reader driver whether or not a voltage selection shall be performed or not. If set to REQ_VOLTAGE_SEL, the SIM reader driver does a regular reset of the card in which it performs the voltage selection procedure.

If set to OMIT_VOLTAGE_SEL and the card has been activated with a simdrv_reset() call prior to this, thus having a used voltage already, it shall reset the card using only the currently used voltage. This also applies for *UICC Characteristics*.

This is a special case, where the SIM application toolkit requests a reset of the card, see [3G 11.14, 6.4.7]

Definition:

Type	Short Name	Comment	Group
U8	voltage_select	Perform Voltage Selection	uicc

Values:

Value	C-Macro	Comment
0	REQ_VOLTAGE_SEL	The SIM reader driver shall perform voltage selection
1	OMIT_VOLTAGE_SEL	The SIM reader driver shall restart with the current voltage

History:

03-May-04 FDU Initial

5.13 SIM Command Parameters

Description:

The parameters P1 and P2 are individually coded in relevance to what type of command is to be performed. Coding according to [3G 11.11] and [ETSI 102 221]

Definition:

Type	Short Name	Comment	Group
U8	p1	Parameter 1 of the SIM APDU	uicc
U8	p2	Parameter 2 of the SIM APDU	uicc

History:

03-May-04 FDU function renamed, parameters aligned to review report

5.14 Configuration Characteristics Request

Description:

This parameter is used by the driver to indicate that the PS must retrieve UICC characteristics from the MF (‘3F00’) and deliver them to the driver using the provided pointer reference (see chapter

5.15). This will occur in situations were the inserted cards does not deliver e.g. *preferred clock stop level* in the ATR. If the value of the parameter is SIMDRV_OMIT_CONFIG_CHARACTERISTICS, the PS must never try to access data using the provided pointer.

Definition:

Type	Short Name	Comment	Group
U8	config_requested	Configuration Characteristics requested	uicc

Values:

Value	C-Macro	Comment
0x00	OMIT_CONFIG_CHARACTERISTICS	PS Shall not retrieve Configuration Characteristics
0x01	REQUEST_CONFIG_CHARACTERISTICS	PS Shall retrieve Configuration Characteristics

History:

10-June-04 FDU function renamed, parameters aligned to review report

5.15 Configuration Characteristics

Description:

The PS shall deliver the configuration characteristics coded according to [ETSI 102 221, 11.1.1.4.6.1].

Definition:

Type	Short Name	Comment	Group
STRUCT	config_characteristics	Configuration Characteristics	uicc

Elements:

Long Name	Short Name	CTRL	Ref	Type
UICC Characteristics	uicc_characteristics		5.17	U8

History:

10-June-04 FDU function renamed, parameters aligned to review report

5.16 Reader Id

Description:

Used to indicate to the driver which reader the currently command is issued towards.

Definition:

Type	Short Name	Comment	Group
U8	reader_id	Reader Id	uicc

Values:

Value	C-Macro	Comment
0x01-MAX_READERS		Range of Id's of readers

History:

10-June-04 FDU function renamed, parameters aligned to review report

5.17 UICC Characteristics

Description:

Used to deliver the UICC Characteristics of the MF to the driver

Definition:

Type	Short Name	Comment	Group
U8	uicc_characteristics	UICC Characteristics	uicc

History:

10-June-04 FDU function renamed, parameters aligned to review report

/*

6 MSCs

6.1 Successful activation

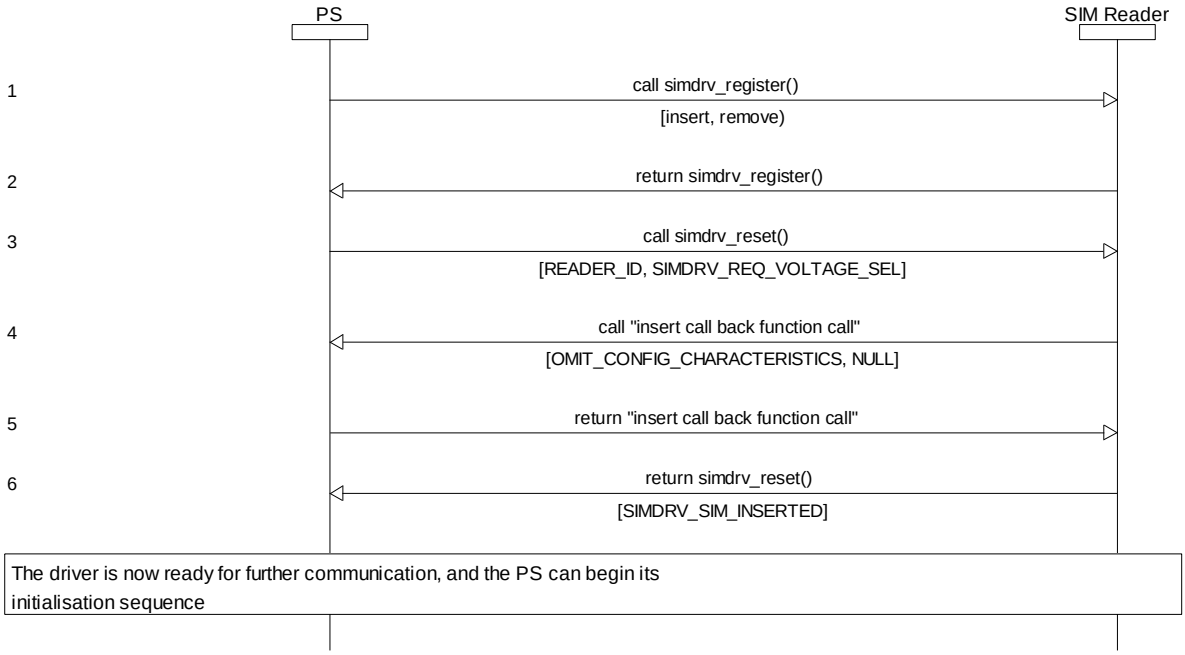

Figure 2 - Initial communication, successful activation

Figure 2 shows the initial communication between the PS and the driver. When the PS is initialised it registers the driver with the simdrv_register function. In this function call the driver is handed the two function pointers for respectively insert and remove functions.

When the PS is activated or wants to reset the inserted UICC it uses the simdrv_reset function to request the driver to attempt to reset the inserted UICC.

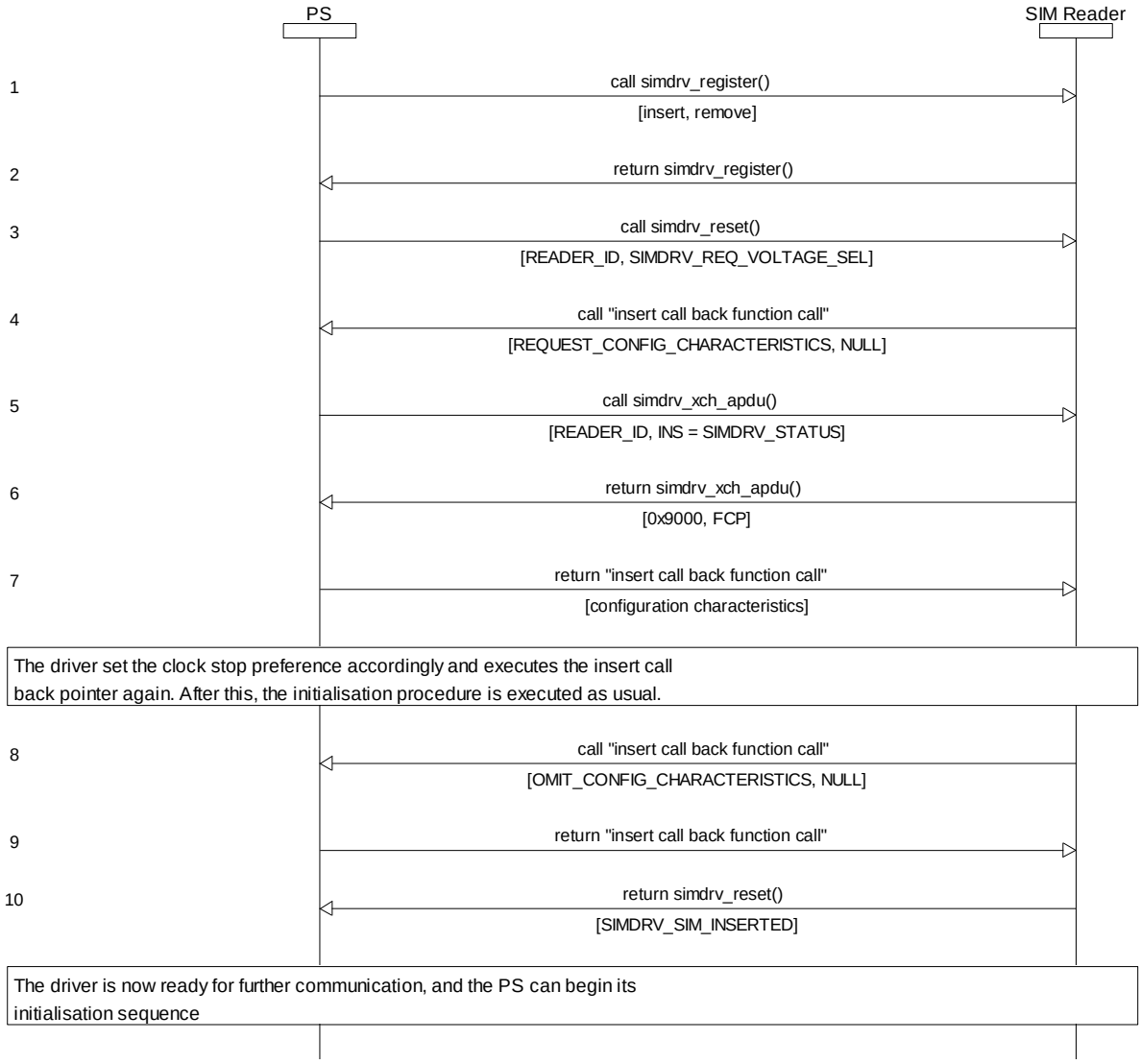

Figure 3 - Initial communication, successful activation, configuration information required.

If the reset was successful, the driver uses the insert function pointer to execute the PS's initialisation procedure (an example can be found on Figure 2).

In cases where the inserted UICC does not return any UICC characteristics, e.g. clock stop preferences as a part of the ATR, the driver will request these settings from the PS. This sequence will be that shown on Figure 3.

6.2 Unsuccessful activation

If the reset fails, the driver returns a error code other than SIM_INSERTED and does not execute the insert function pointer.(an example can be found on Figure 4) .

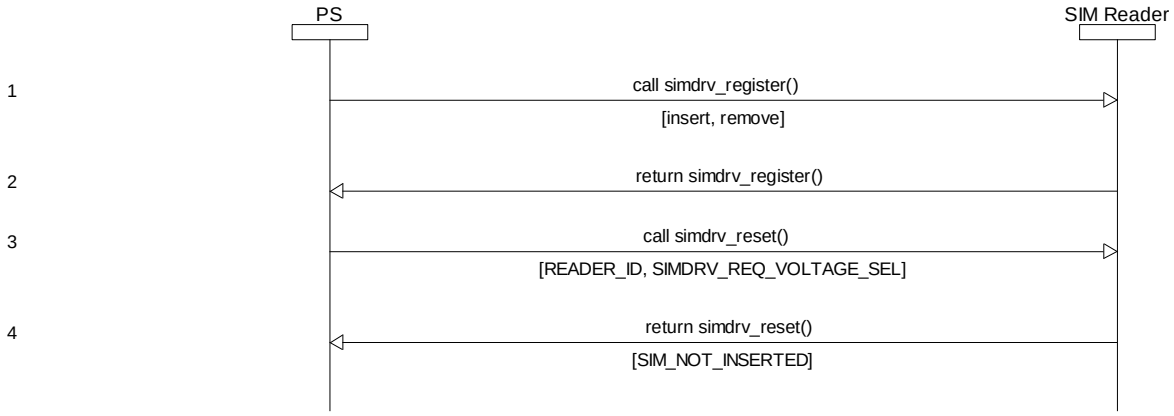

Figure 4 - Unsuccessful initial communication

6.3 Power off

At any time, the PS can perform a power off on the driver. This power off shall be performed according to [3G 11.11] and [ETSI102 221].

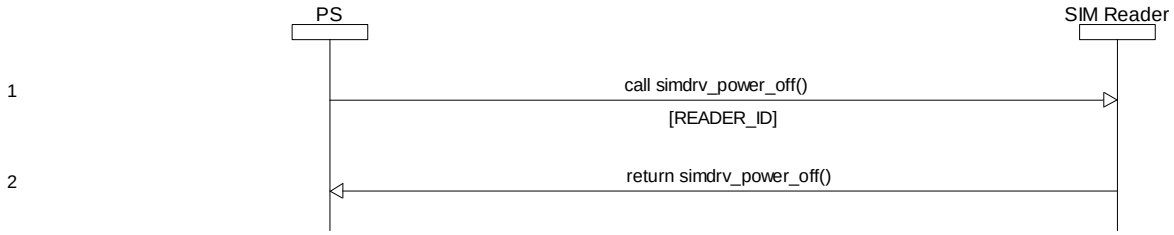

Figure 5 - Powering off

6.4 Issuing commands

The PS uses the simdrv_xch_apdu() function to issue commands towards the reader with the Id given.

If the reader with the given Id has not been reset, a proper error code shall be returned and the driver shall not issue the requested command.

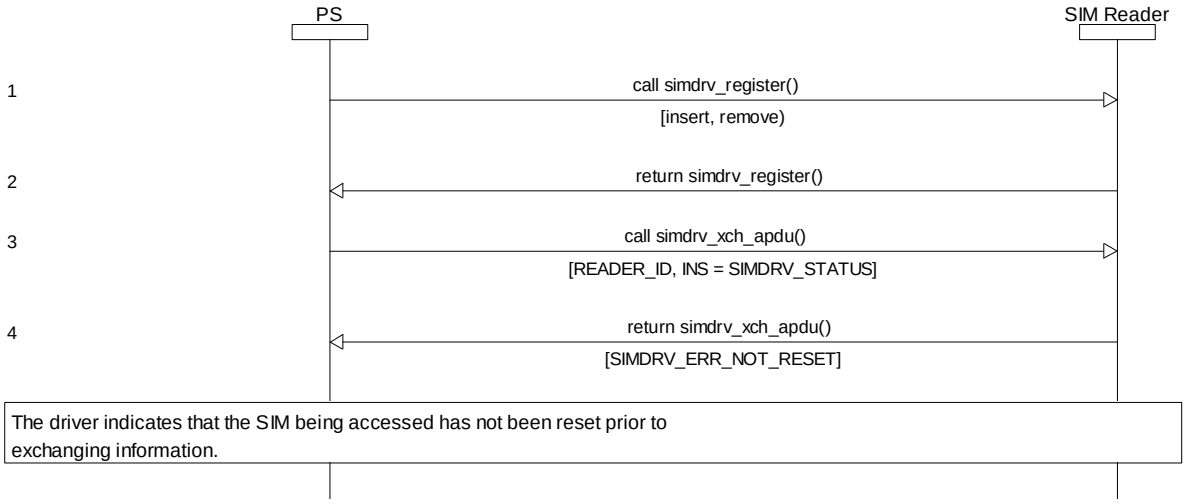

Figure 6 - Unsuccessful communication, missing reset

6.5 Command Types

The [ETSI 102 221] describes 4 different types of commands. As the access protocol is handled autonomously by the PS, one more case appears, case 5.

1. No sending data, no receiving data.
2. Sending data, no receiving data.
3. No sending data, receiving data.
4. Sending data, receiving data
5. Sending/No sending of data, receiving data of unknown length

Each of the five cases is exemplified below.

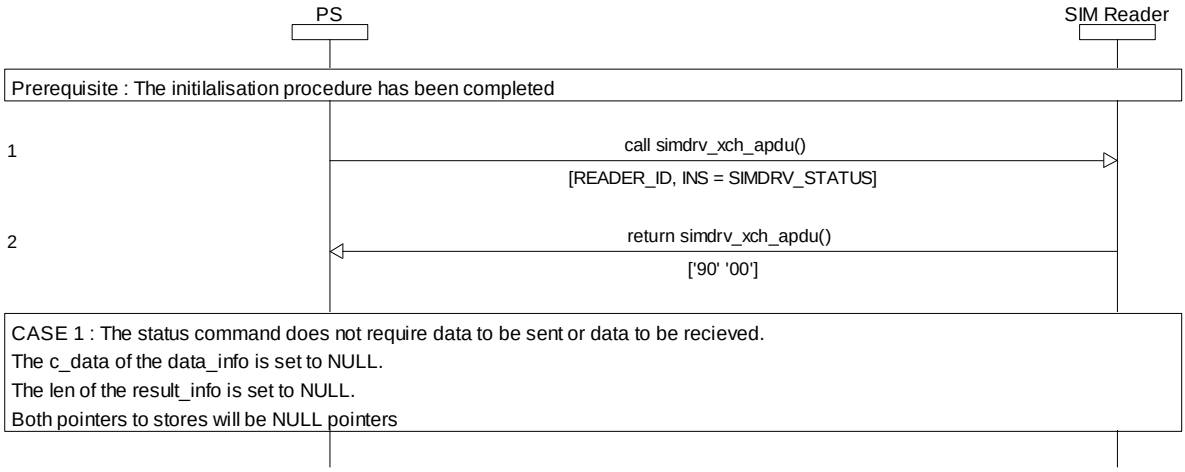

Figure 7 - Case 1 command example

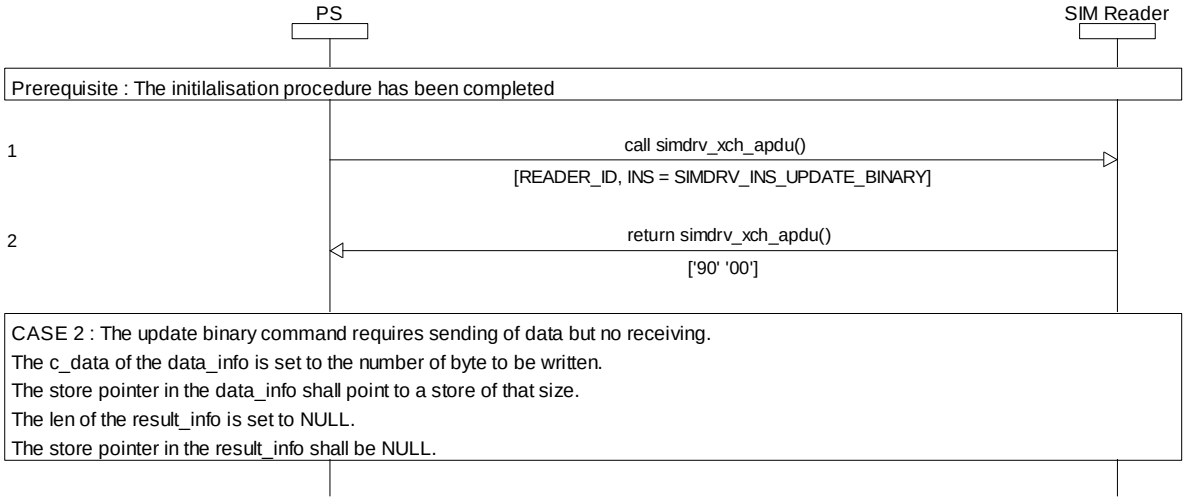

Figure 8 - Case 2 command example

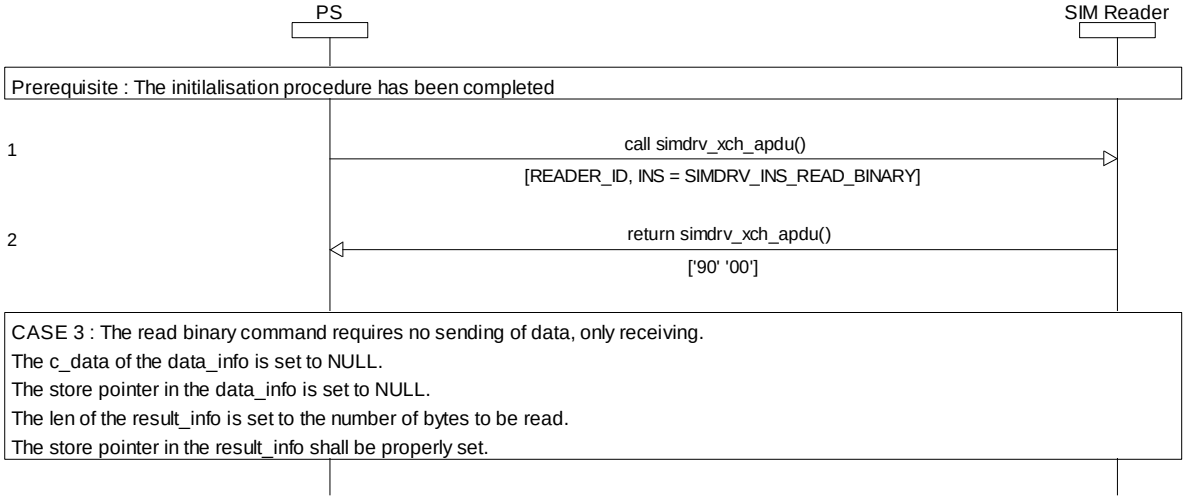

Figure 9 - Case 3 command example

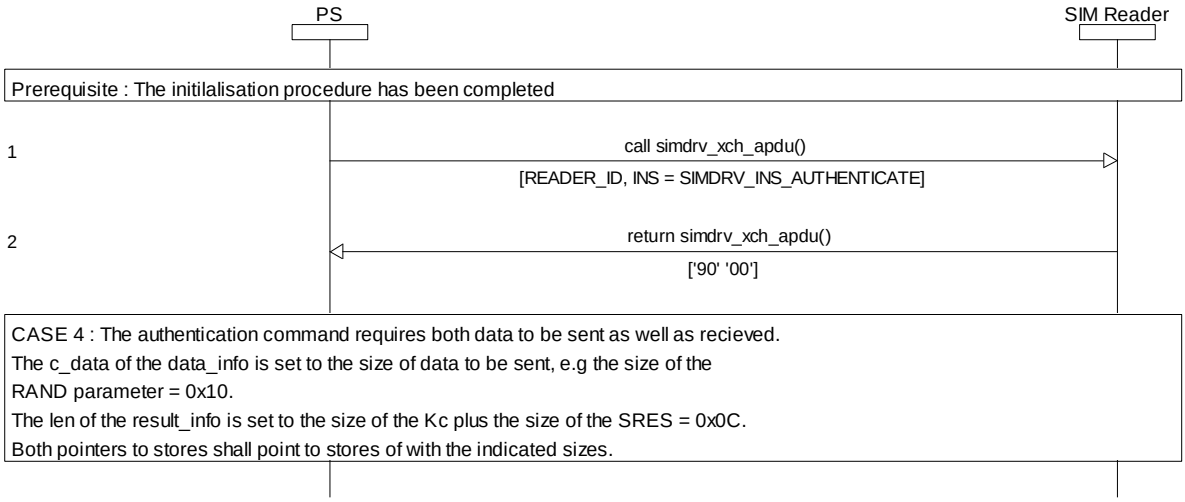

Figure 10 - Case 4 command example

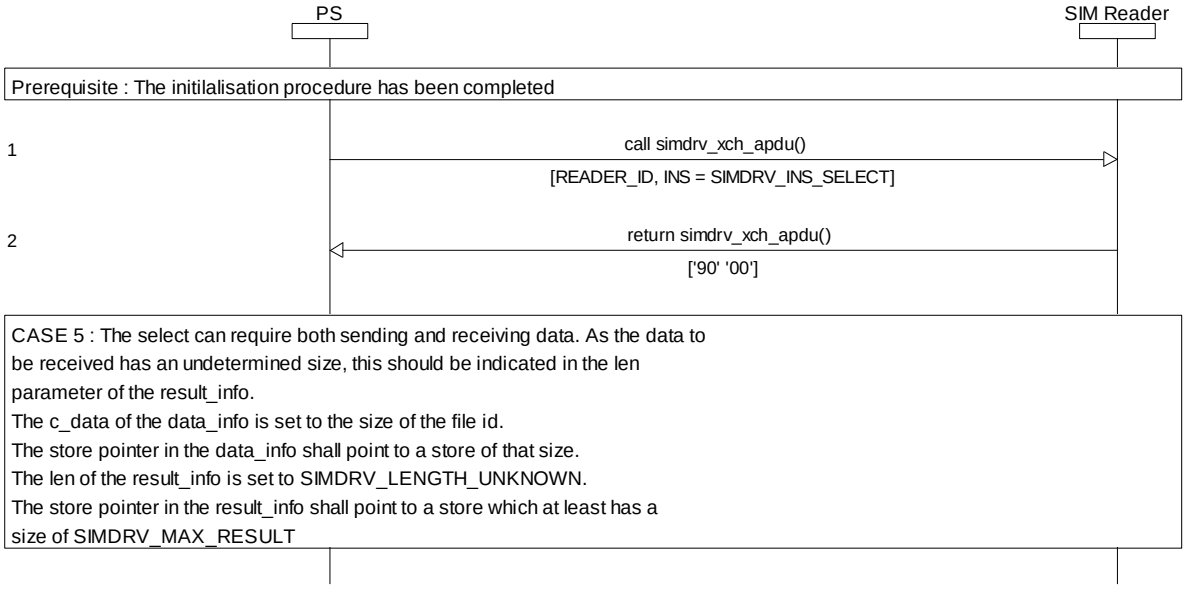

Figure 11 - Case 5 command example

*/