Technical Document - Confidential

GSM PROTOCOL STACK

GPF

TCSL – TEST CASE SCRIPT LANGUAGE

USER GUIDE

Document Number:	06-03-30-UDO-0000
Version:	0.6
Status:	Draft
Approval Authority:	
Creation Date:	1997-Nov-27
Last changed:	2015-Mar-08 by XINTEGRA
File Name:	6147_302.doc

Change History

Date	Changed by	Approved by	Version	Status	Notes
1997-Nov-27	VK et al.		0.1		1
1997-Nov-27	VK et al.		0.2		2
1999-Mar-30	MS et al.		0.3		3
2000-Mar-31	HSC et al.		0.4	Status	4
2003-May-13	XINTEGRA		0.5	Draft	
2003-Sep-08	HSC		0.6		5

Notes:

1. Initial version
2. ACT_SHOW, MSG3 has Frame Number, MSG3_AWAIT_COPY
3. New format/English check
4. New Macros, Anite-Adaption
5. New document number

Table of Contents

List of Figures and Tables

List of References

[ISO 9000:2000]	International Organization for Standardization. Quality management systems - Fundamentals and vocabulary. December 2000

1 Introduction

G23 is a software package implementing Layers 2 and 3 of the ETSI-defined GSM air interface signaling protocol, and as such represents the part of a GSM mobile station's protocol software which is both, platform and manufacturer independent. Therefore, G23 can be viewed as a building block providing standardized functionality through generic interfaces for easy integration.

The G23 suite of products consists of the following items:

- Layers 2 and 3 for speech & short message services,
- Layers 2 and 3 for fax & data services,
- Application Control Interface,
- Slim MMI [02.30] and
- Test and integration support tools.

This document describes TCSL.

TCSL:

- is an acronym for "Test Case Script Language".
- is a notation for the description of GSM conformance tests according to the GSM 11.10 specification.
- allows a notation of the GSM 11.10 text in a C-preprocessor readable form.
- is a collection of C-preprocessor function-like macros.
- may be adapted to other, non-GSM protocols.
- can be executed on a system which implements a Layer 2 of the GSM infrastructure.
- can be adapted to radio testers.

2 TCSL structure

TCSL consists of two parts:
- a static part containing data
- a dynamic part containing code

2.1 TCSL static part

The static part of TCSL defines three types of data items:
- bit fields (BF)
- information elements (IE)
- Layer 3 messages (MSG3)

The following six macros are used to define bit fields, information elements and Layer 3 messages :

- IE_BEGIN	start of the definition of an information element
- BF	definition of a bit field
- IE_END	end of the definition of an information element
- MSG3_BEGIN	start of the definition of a Layer 3 message
- IE	reference to an information element
- MSG3_END	end of the definition of a Layer 3 message

The macro BF may appear only between IE_BEGIN and IE_END.
The macro IE may appear only between MSG3_BEGIN and MSG3_END.

A hierarchy of the data types exists :
A Layer 3 message is a sequence of information elements
An information element is a sequence of bit fields.

2.1.1 Bit fields

Bit fields are the basic data items of TCSL. A bit field is defined with the macro BF. A bit field may be defined only during the definition of an information element (between IE_BEGIN and IE_END). The attributes of a bit field are:

- length	(const)
- value	(var)
- action	(var)
- name	(const)
- comment	(var)

These attributes are given to the macro BF as parameters:
BF (LEN, VAL, ACT, NAM, COM)

Once defined, the length and the name of a bit field cannot be changed. The value of the assigned action, comment and value may be changed with TCSL macros.

LEN :
A bit field has a fixed length of 1 to 32 bits.

VAL:
Depending on its size, a bit field can take on values between 0 and 4294967295. Any valid C expression producing a value of integral type may be used as VAL in a call to BF. The "LEN" least significant bits of "VAL" are used to initialize the bit field. Here "least significant" means, that the expression
(VAL & ((1<<LEN) - 1)) masks out the those bits which are used for the initial value of the bit field (assuming an unrestricted operator << on the constant "1").

ACT:
The "action" attribute of a bit field serves as a link to the dynamic part of TCSL. The action can be:

ACT_NOP	no operation should be performed on the bit field
ACT_CHECK	the defined bit field contains the expected value of a received bit field
ACT_SHOW	similar to ACT_NOP, but after an AWAIT operation the received bit field is outputted
ACT_FILL	the bit field is not used during SEND and AWAIT operations
ACT_STOP	the bit field has not been fully specified, stop the test immediately

The value ACT_FILL can be thought of as an "off" - switch for the bit field. Its primary use is to define optional bit fields.

NAM:
A bit field may be assigned a name. It should be a valid name as defined in the programming language C. The special name ANONYMOUS can be used if no name should be assigned to a bit field. The name of a bit field is used when the value of the bit field is changed with the macros BF_SET_VAL and MSG3_BF_SET_VAL (in the dynamic part of TCSL). The name of a bit field is specified without quotation marks.

COM:
A comment may be assigned to a bit field. If no comment is used, then the special comment SILENT must be specified. A comment must be specified in quotation marks ("").

Here are some examples of bit fields:
```
BF (  1,       0, ACT_NOP  ,    ANONYM ,                SILENT )
BF ( 23, 0x12345, ACT_CHECK, a_bit_field, "example of a bit field")
```

2.1.2 Information elements

An information element is basically a non-empty sequence of bit fields. An information element is defined with the macros IE_BEGIN and IE_END which surround the definition (using macro BF) of those bit fields belonging to the information element. Aside from the BitFields, the only attribute of an information element is its name which is given to the macro IE_BEGIN and IE_END as a parameter:

IE_BEGIN (NAME)
IE_END (NAME)

The name of an information element is specified without quotation marks.

Here is an example of the definition of an information element:

```
IE_BEGIN  ( rach_control_parameter )
       BF (  2, 0, ACT_NOP, max_retrans, "Maximum number of retransmissions" )
       BF (  4, 0, ACT_NOP,  tx_integer, "Number of slots to spread transmission" )
       BF (  1, 0, ACT_NOP,         cba, "Cell Bar Access" )
       BF (  1, 1, ACT_NOP,          re, "Call Re-establishment" )
       BF (  5, 0, ACT_NOP,    ac_15_11, "Access Control class 15 through 11" )
       BF (  1, 1, ACT_NOP,          ec, "Emergency Call" )
       BF ( 10, 0, ACT_NOP,    ac_09_00, "Access Control class 9 through 0" )
IE_END    ( rach_control_parameter )
```

2.1.3 Layer 3 messages

2.1.3.1 "Normal", initialized Layer 3 messages

A Layer 3 message is basically a non-empty sequence of information elements. A Layer 3 message is defined with the macros MSG3_BEGIN and MSG3_END which surround references (using macro IE) to those information elements belonging to the Layer 3 message. Aside from the information elements, the attributes of a Layer 3 message are (1) a time indication and (2) the name of the message which is given to the macro MSG3_BEGIN and MSG3_END as a parameter:

MSG3_BEGIN (NAME)
MSG3_END (NAME)

The name of a Layer 3 message is specified without quotation marks.
The time indication has been introduced in TCSL, version 1.1. The time indication is a frame number captured at the reception of a message. It should be noted that the frame number is the only variable part of a MSG3 object.

Here is an example of a definition of a Layer 3 message:

```
MSG3_BEGIN(system_information_type_1)
   IE(l2_pseudo_length_21)
   IE(rr_management_protocol_discriminator)
   IE(skip_indicator)
   IE(system_information_type_1_message_type)
   IE(cell_channel_description)
   IE(rach_control_parameter)
   IE(si_1_rest_octets)
MSG3_END(system_information_type_1)
```

This example does not contain the definition of the information elements. The definition of `rach_control_parameter` is provided in the previous section.

2.1.3.2 Empty Layer 3 messages

Two types of Layer 3 messages exist : initialized and non-initialized Layer 3 messages. The initialized Layer 3 message is the "normal" case. An initialized Layer 3 messages is constant (except the Frame

Number). Its contents cannot be changed (except the Frame Number). In order to create a new Layer 3 message in terms of an existing Layer 3 message, an empty Layer 3 message can be defined (in the static part of TCSL) and assigned a value (in the dynamic part of TCSL) from the initialized Layer 3 message. The macro MSG3_VAR is used to define an empty (non-initialized) Layer 3 message. The macro MSG3_COPY_CREATE is used to copy the values of an initialized Layer 3 message to an empty Layer 3 message. When the Layer 3 message is no longer used, the macro MSG3_COPY_DESTROY must be applied on that message. Here is an example of how to use MSG3_VAR :

```
MSG3_VAR ( si1_copy )
/* system_information_type_1 definition goes here */

MSG3_COPY_CREATE  ( si1_copy, system_information_type_1 )
MSG3_BF_SET_VAL   ( si1_copy, rach_control_parameter.cba, 1 )

/* use si1_copy here, then destroy it */

MSG3_COPY_DESTROY ( si1_copy )
```

In the example, the existence of a Layer 3 message with the name "system_information_type_1" is assumed. The macros MSG3_COPY_CREATE, MSG3_COPY_DESTROY and MSG3_BF_SET_VAL are described in the following section.

2.2 TCSL dynamic part

The dynamic part of TCSL contains the following macros:

Infrastructure simulator (ISS):		
	ISS_INIT	(NUM_BS)
	ISS_DEINIT	()
	ISS_DELAY	(MILLI_SEC)
Base station (BS) configuration:		
	BS_SET_SYS_INFO	(BS_IDX,SI)
	BS_RESET_SYS_INFO	(BS_IDX,SI)
	BS_ON_OFF	(BS_IDX,ON_OFF)
	BS_SET_POWER	(BS_IDX,POWER)
	BS_SET_ARFCN	(BS_IDX,ARFCN)
	BS_SET_SCH	(BS_IDX,BSIC,RFN)
	BS_CONFIG_CHANNEL	(BS_IDX,CHANNEL,TYPE,SAPI)
	BS_CONFIG_FROM_FILE	(FILE)
Send/Await Layer 3 messages:		
	BS_MSG3_SEND	(BS_IDX,MSG3,COM)
	BS_MSG3_AWAIT	(BS_IDX,MSG3,COM)
	BS_RACH_AWAIT	(BS_IDX,RACH,COM)
	BS_MSG3_AWAIT_COPY	(BS_IDX,MSG3,MSG3_COPY,COM)
	BS_RACH_AWAIT_COPY	(BS_IDX,RACH,RACH_COPY,COM)
	BS_MSG3_SEND_BEGIN	(BS_IDX,MSG3,MSG3_COM)
	BS_MSG3_AWAIT_BEGIN	(BS_IDX,MSG3,MSG3_COM)
	BS_RACH_AWAIT_BEGIN	(BS_IDX,MSG3,RACH_COM)
	BF_SET_VAL	(BF_NAM,BF_VAL,BF_COM)
	BS_RACH_AWAIT_END	()
	BS_MSG3_AWAIT_END	()
	BS_MSG3_SEND_END	()

Layer 3 message manipulation:

MSG3_COPY_CREATE	(MSG3_COPY,MSG3)
MSG3_COPY_DESTROY	(MSG3_COPY)
MSG3_BF_SET_VAL	(MSG3,BF_NAM,BF_VAL,BF_COM)
MSG3_BF_GET_VAL	(MSG3,BF_NAM)
MSG3_BF_SET_ACT	(MSG3,BF_NAM,BF_ACT)
MSG3_BF_GET_ACT	(MSG3,BF_NAM)

Miscellaneous:

NOT_IMPLEMENTED	(COMMENT)
SET_TIMEOUT	(MILLI_SEC)
DELAY	(MILLI_SEC)
BS_SET_ERROR	(BS_IDX,COUNT)
BS_STORE_RACH_PARAMS	(BS_IDX,POS)
USER_INFO_BOX	(ANNOUNCEMENT)

The macros have been categorized into five groups. However, some macros belongs to more than one group. These macro groups are described in the following section.

2.2.1 TCSL macros related to "Infrastructure Simulator (ISS)"

ISS_INIT	(NUM_BS)
ISS_DEINIT	()
ISS_DELAY	(MILLI_SEC)

The execution of a test case written in TCSL is based on a system called the infrastructure simulator (ISS). In order to initialize the ISS, the first macro of a test case should be ISS_INIT. This macro has one parameter "NUM_BS" which is the number of base stations that should be simulated in the infrastructure simulator. Initially all requested base stations are "off" but otherwise do not have a default state (e.g. System Information messages, BSIC) assigned.
The last macro of a test case in TCSL should be ISS_DEINIT which does not have any parameters. However, an opening and a closing parentheses must be supplied. The macro ISS_DEINIT puts the infrastructure simulator into a state similar to the state before ISS_INIT was issued.
The macro ISS_DELAY forces the ISS to wait for specified amount of time. During this time, no Layer 3 messages (exceptions: system information messages and measurement reports) are sent or received by the ISS.

2.2.2 TCSL macros related to "Base Station (BS) Configuration"

BS_SET_SYS_INFO	(BS_IDX,SI)
BS_RESET_SYS_INFO	(BS_IDX,SI)
BS_ON_OFF	(BS_IDX,ON_OFF)
BS_SET_POWER	(BS_IDX,POWER)
BS_SET_ARFCN	(BS_IDX,ARFCN)
BS_SET_SCH	(BS_IDX,BSIC,RFN)
BS_CONFIG_CHANNEL	(BS_IDX,CHANNEL,TYPE,SAPI)

The macros of the category "Base Station (BS) Configuration" are used to set the state of a base station of the infrastructure simulator. This state is summarized by the following attributes:

- system information messages (SI)
- BS switched on or off (ON_OFF)
- power in dBm (POWER)
- Absolute Radio Frequency Channel Number (ARFCN)
- Base Station Identification Code and Reduced Frame Number (BSIC and RFN)
- channel configuration (CHANNEL, TYPE, SAPI)

The macros are used to set these attributes. There are no macros to retrieve the value of these attributes.
The parameter BS_IDX specifies the index of the base station within the infrastructure simulator. This index can take on values 0 <= BS_IDX <= NUM_BS-1 (NUM_BS was specified in a call to ISS_INIT).

2.2.3 TCSL macros related to "Send/Await Layer 3 messages"

The "Send/Await" macros are categorized into "basic" macros and macros which allow the assignment of values to bit fields immediately before the SEND or AWAIT operation is performed.

2.2.3.1 Basic SEND/AWAIT

```
BS_MSG3_SEND              (BS_IDX,MSG3,COM)
BS_MSG3_AWAIT             (BS_IDX,MSG3,COM)
BS_RACH_AWAIT             (BS_IDX,RACH,COM)
```

The macros in the category "basic Send/Await" are used to exchange Layer 3 messages between the ISS and the IUT. As a side effect, a received Layer 3 message is compared with the expected Layer 3 message and the result on this comparison is outputted on the computer screen and in a data file. More precisely: only those bit fields of the messages which have the ACT_CHECK attribute set are compared.
The two basic macros of the "Send/Await" category are BS_MSG3_SEND and BS_MSG3_AWAIT. Both macros have three parameters BS_IDX, MSG3 and COM. The parameter BS_IDX is the index to the base station within the infrastructure simulation on which the SEND or AWAIT operation should be performed. The parameter BS_IDX may take on values between 0 and NUM_BS-1 (where NUM_BS has been specified in a call to ISS_INIT). The parameter MSG3 describes the Layer 3 message that should be sent. In case of an AWAIT action MSG3 describes the expected Layer 3 message.
The values of the "action" attribute of the bit fields defined in the information elements of the Layer 3 messages control the behavior of the SEND and AWAIT macros :

	SEND	**AWAIT**
ACT_NOP	send bit field	receive bit field, but do not compare
ACT_CHECK	same as ACT_NOP	receive bit field and compare
ACT_FILL	bit field not sent	no bit field received and compared
ACT_STOP	stop test case	stop test case
ACT_EOIE	end of IE	end of IE

The macro BS_RACH_AWAIT is similar to BS_MSG3_AWAIT, except that BS_RACH_AWAIT is applicable for a special type of Layer 3 message only, the "channel request message" (maybe we can get rid of BS_RACH_AWAIT if the macro BS_CONFIG_CHANNEL can be used to specify the RACH).

The third parameter of the macro is a comment describing the operation performed. This comment must be specified in quotation marks.

2.2.3.2 SEND/AWAIT with bit field assignment

```
BS_MSG3_SEND_BEGIN        (BS_IDX,MSG3,MSG3_COM)
BS_MSG3_AWAIT_BEGIN       (BS_IDX,MSG3,MSG3_COM)
BS_RACH_AWAIT_BEGIN       (BS_IDX,MSG3,RACH_COM)

BF_SET_VAL          (BF_NAM,BF_VAL,BF_COM)

BS_RACH_AWAIT_END         ()
```

BS_MSG3_AWAIT_END	()
BS_MSG3_SEND_END	()

The macros of the category "SEND/AWAIT with bit field assignment" serve the same purpose as the basic SEND/AWAIT macros with the additional feature, that individual bit fields may be altered when the SEND or AWAIT operation is performed. For this purpose a complete SEND operation consists of
- one call to BS_MSG3_SEND_BEGIN followed by
- one or more calls to BF_SET_VAL followed by
- one call to BS_MSG3_SEND_END

An AWAIT operation (either Layer 3 or RACH) has a similar structure.
Here is an example :

```
IE_BEGIN(introduction)
       BF(12,    0x007,ACT_NOP,     bond,"the double '0' agent")
IE_END(introduction)

IE_BEGIN(my_name_is)
       BF(28,0xAAAAAAA,ACT_NOP,     james,"here I am"           )
       BF( 8,     0x77,ACT_NOP,      bond,"seventy-seven sunset" )
       BF(16,   0xDEAD,ACT_NOP,in_the_bar,"shaken, not stirred " )
IE_END(my_name_is)

MSG3_BEGIN(new_movie)
       IE(introduction)
       IE(my_name_is)
MSG3_END(new_movie)

BS_MSG3_AWAIT_BEGIN(0,new_movie,"1:the best bond ever")
       BF_SET_VAL (       in_the_bar, 0x1234, SILENT )
       BF_SET_VAL ( introduction.bond,  0x000, SILENT )
BS_MSG3_AWAIT_END()
```

In this example the Layer 3 message "new_movie" is received at base station 0. Immediately before the AWAIT operation is performed a temporary copy of the original Layer 3 message is made and two bit fields in that copy are altered. The first bit field altered is "in_the_bar" which is assigned the new value "0x1234". The second bit field altered is "bond" in the information element "introduction". It should be noted that the bit field name "bond" appears twice in the Layer 3 message. Therefore, the prefix "introduction." is required. After the SEND or AWAIT operation has been performed, the temporary copy of the Layer 3 message is destroyed. The modification of bit fields cannot be performed on ANONYMOUS bit fields.

There are two important side effects when using BF_SET_VAL:
Within an AWAIT_BEGIN/END operation, the altered bit field is assigned the action value ACT_COMPARE.
Within a SEND_BEGIN/END operation, the altered bit field is assigned the action value ACT_NOP.

The mechanism to change the value (and comment) of bits fields when the AWAIT or SEND operation is performed may reduce the amount of data in the static part of TCSL. However it increases the number of lines in the dynamic part of TCSL.

2.2.4 TCSL macros related to "Layer 3 message manipulation"

MSG3_COPY_CREATE	(MSG3_COPY,MSG3)
MSG3_COPY_DESTROY	(MSG3_COPY)
MSG3_BF_SET_VAL	(MSG3,BF_NAM,BF_VAL,BF_COM)
MSG3_BF_GET_VAL	(MSG3,BF_NAM)
MSG3_BF_SET_ACT	(MSG3,BF_NAM,BF_ACT)
MSG3_BF_GET_ACT	(MSG3,BF_NAM)

The macros of the category "Layer 3 message manipulation" serve the following purposes :
- create and destroy copies of Layer 3 messages

- changed the value of bit fields
- retrieve the value of bit fields

The macro MSG3_COPY_CREATE is used to create a copy of a Layer 3 message. The first parameter MSG3_COPY of the macro MSG3_COPY_CREATE is the "destination" and second parameter MSG3 is the source of this copy operation. All elements of the Layer 3 message are copied including those bit fields which have the "action" attribute set to ACT_FILL. The copy process is "flat" (addresses are copied) with the exception of those elements which may be altered by a MSG3_BF_SET operation. Thus, the original Layer 3 message (source) is not changed by a MSG3_BF_SET operation.

The macro MSG3_COPY_DESTROY is used to destroy a copy of a Layer 3 message previously created by MSG3_COPY_CREATE.

Important: the first parameter of the MSG3_COPY_CREATE and MSG3_COPY_DESTROY macros must be defined by the macro MGS3_VAR (i.e. MSG3_COPY must be an "empty" Layer 3 message).

The four SET/GET macros are used to change and to retrieve the value, comment and action assigned of the specified bit field. The bit field is identified by its name as specified in the definition of the bit field with macro BF. If however the same bit field name exists more than once within the specified message, the bit field name must be dot-prefixed by the name of the information element containing the bit field. If the name is not unique or does not exist, the test case result is "not passed".

Here is an example using the Layer 3 message "new_movie" from the previous section:

```
MSG_EMPTY(part_2)

MSG3_COPY_CREATE(part_2,new_movie)

MSG3_BF_SET_VAL (part_2,        in_the_bar, 0x1234, SILENT )
MSG3_BF_SET_VAL (part_2, introduction.bond,  0x000, SILENT )

BS_MSG3_AWAIT(0,part_2,"1:the best bond ever")

MSG3_COPY_DESTROY(part_2)
```

The example performs the same operations on the ISS/IUT interface as in the example previous section. The difference is, that the temporary copy of the Layer 3 message is now managed explicitly in TCSL and named "part_2".

2.2.5 TCSL unrelated macros, "Miscellaneous"

```
NOT_IMPLEMENTED  (COMMENT)
SET_TIMEOUT      (MILLI_SEC)
BS_SET_ERROR             (BS_IDX,COUNT)
BS_STORE_RACH_PARAMS (BS_IDX,POS)
```

A few TCSL macros exist which are difficult to assign to a special category. These macros are described in this section.

The macro NOT_IMPLEMENTED should be used to signal those parts of the test case specification which cannot be implemented in TCSL. During the execution of a TCSL test case a comment is outputted to the computer screen as a warning that the test case is not fully implemented.

The SET_TIMEOUT macro is used to set the maximum amount of time in milliseconds, which the ISS should wait when performing an AWAIT operation. If this time passes by without receiving a message, the test case is stopped with a "not passed" result.

The BS_SET_ERROR macro specifies the number of bad frames introduced by the ISS to the IUT as a stimulus.

The BS_STORE_RACH_PARAMS is used to save the value of the RACH message (channel request message) for use in the IMMEDIATE ASSIGMENT (or similar) message.

3 TCSL macros

In this section all TCSL macros are described. A description, category, syntax, example and reference to other macros are provided for each macro.

3.1 BF

Description: The BF macro is used to define bit fields. The macro BF may only appear between IE_BEGIN and IE_END.

Category: Data Definition

Syntax: BF (LEN , VAL , ACT , NAM , COM)

LEN: The number of bits in the bit field (1<=LEN<=32).
The length may be specified as an expression of the programming language C which
produces a constant value.
VAL: The initial value of the bit field.
The least significant LEN bits of VAL are used to initialize the bit field.
VAL may be any expression of the programming language C which produces a constant value.
ACT: The action assigned to the bit field. It specifies an operation when a SEND or AWAIT is performed. ACT may be:

ACT_NOP	: no operation
ACT_CHECK	: expected and received messages are compared in AWAIT operations
ACT_FILL	: the bit field does not take part in SEND or AWAIT operations
ACT_STOP	: the bit field is not fully defined, the test case is stopped
ACT_EOIE	: end of information element marker

NAM: The name of the bit field. The coding convention is: use names as in the programming language C consisting of lower case letters and the underline char "_".
There is no restriction on the length of the name.
The special name ANONYMOUS may be used.
COM: A comment describing the bit field in more detail.
The comment must be specified in quotation marks.
There is no restriction on the length of the comment.
The special comment SILENT may be used.

Examples:

```
IE_BEGIN ( a_sample_ie )
   BF ( 4, 0x04, ACT_NOP, ANONYMOUS, SILENT )
IE_END ( a_sample_ie )

IE_BEGIN ( another_ie_10_bits_long )
   BF ( 1+2+3+4, 0, ACT_CHECK, what_is_the_word, "ten bits : 1+2+3+4" )
IE_END ( another_ie_10_bits_long  )
```

See also: IE_BEGIN, IE_END

3.2 BF_SET_VAL

Description: The macro BF_SET_VAL is used to change the value of a bit field immediately before a SEND or AWAIT operation is performed. The macro BF_SET_VAL may appear between BS_MSG3_SEND_BEGIN and BS_MSG3_SEND_END, between BS_MSG3_AWAIT_BEGIN and BS_MSG3_AWAIT_END or between BS_RACH_AWAIT_BEGIN and BS_RACH_AWAIT_END.

Category: Send/Await Layer 3 messages

Syntax: BF_SET_VAL (BF_NAM , BF_VAL , BF_COM)

BF_NAM :	The name of the bit field as defined in the corresponding BF macro. If the name occurs more than once in the Layer 3 message that should be sent or received, then the name of the bit field must be prefixed by the name of the information element containing the bit field. In this case, the name of the information element and the name of the BitField must be separate by a dot.
BF_VAL:	The new value of the bit field. Refer to the description of "VAL" in "macro BF" for details.
BF_COM:	A comment describing the value of the bit field. The comment must be specified in quotation marks. There is no restriction on the length of the comment. The special comment SILENT may be used.

Example:
```
BS_MSG3_SEND_BEGIN ( 0, a_message )
   BF_SET_VAL ( ie2.bf5, 0x1234, "assign 0x1234 to the bf5 in info element ie2" )
BS_MSG3_SEND_END()
```

See also: BS_MSG3_SEND_BEGIN,
BS_MSG3_AWAIT_BEGIN
BS_RACH_AWAIT_BEGIN
BS_MSG3_SEND_END
BS_MSG3_AWAIT_END
BS_RACH_AWAIT_END

3.3 BS_CONFIG_CHANNEL

Description: The macro BS_CONFIG_CHANNEL is used to configure the channel of a base station within the infrastructure simulator.

Category: Base station configuration

Syntax: BS_CONFIG_CHANNEL (BS_IDX, CHANNEL, TYPE, SAPI)

BS_IDX:	Index to a base station within the infrastructure simulator.
CHANNEL:	Either PCH or AGCH or SDCCH or SACCH or FACCH.
TYPE:	One of UNACK or ACK
SAPI:	SAPI_0 or SAPI_3

Example: BS_CONFIG_CHANNEL (0, PCH, UNACK, SAPI_0)

3.4 BS_CONFIG_FROM_FILE

Description: The macro BS_CONFIG_FROM_FILE is used to configure various parameters of various transceivers of a base station. The parameters are read from the given file. Possible parameters are channel combination, time slot, and frequency hopping. An example for a configuration file is given in paragraph 4.5.

Category: Base station configuration

Syntax: BS_CONFIG_FROM_FILE (FILE)

FILE:	Name of the configuration file

Example: BS_CONFIG_FROM_FILE (“anite.cfg”)

3.5 BS_MSG3_AWAIT

Description: The macro BS_MSG3_AWAIT is used to receive a Layer 3 message at the ISS/IUT interface.
After a successful reception of the message, this message is compared with an expected message. If the comparison yields a FALSE result, the test case result is assigned "not passed".
As a side effect of the AWAIT operation, the message is outputted on the computer screen and in a data file.

Category: Send/Await Layer 3 messages

Syntax: BS_MSG3_AWAIT (BS_IDX, MSG3, COM)

BS_IDX:	Index to a base station within the infrastructure simulator.
MSG3:	Expected Layer 3 message.
COM:	A comment describing details about the AWAIT operation. The comment must be specified in quotation marks. There is no restriction on the length of the comment. The special comment SILENT may be used. As coding convention the first part of the comment consists of a number followed by a colon. The number is related to the "step" number of the "expected sequence" in the test cases of GSM 11.10.

Example: BS_MSG3_AWAIT (2, connect, "8:try to receive the CONNECT message")

See also: BS_MSG3_SEND
BS_MSG3_AWAIT_BEGIN
BS_MSG3_AWAIT_END
BS_MSG3_AWAIT_COPY

3.6 BS_MSG3_AWAIT_COPY

Description: The macro BS_MSG3_AWAIT is used to receive a Layer 3 message at the ISS/IUT interface and to copy the received message in a MSG3_VAR object.
After a successful reception of the message, this message is compared with an expected message. If the comparison yields a FALSE result, the test case result is assigned "not passed".
As a side effect of the AWAIT operation, the message is outputted on the computer screen and in a data file.

Category: Send/Await Layer 3 messages

Syntax: BS_MSG3_AWAIT (BS_IDX, MSG3, MSG3_COPY, COM)

BS_IDX:	Index to a base station within the infrastructure simulator.
MSG3:	Expected Layer 3 message.
MSG3_COPY:	The received Layer 3 message.
COM:	A comment describing details of the AWAIT operation. The comment must be specified in quotation marks. There is no restriction on the length of the comment. The special comment SILENT may be used. As coding convention the first part of the comment consists of a number followed by a colon. The number is related to the "step" number of the "expected sequence" in the test cases of GSM 11.10.

Example: BS_MSG3_AWAIT_COPY (2, connect, msg3, "8:try to receive the CONNECT message")

See also: BS_MSG3_SEND
BS_MSG3_AWAIT_BEGIN
BS_MSG3_AWAIT_END

3.7 BS_MSG3_AWAIT_BEGIN

Description: The macro BS_MSG3_AWAIT_BEGIN is used to start the definition of an AWAIT operation. The macro BS_MSG3_AWAIT_BEGIN can only be used in conjunction with BS_MSG3_AWAIT_END.

Category: Send/Await Layer 3 messages

Syntax: BS_MSG3_AWAIT_BEGIN (BS_IDX, MSG3, COM)

BS_IDX:	Index to a base station within the infrastructure simulator.
MSG3:	Expected Layer 3 message.
COM:	A comment describing details about the AWAIT operation. The comment must be specified in quotation marks. There is no restriction on the length of the comment. The special comment SILENT may be used. As coding convention the first part of the comment consists of a number followed by a colon. The number is related to the "step" number of the "expected sequence" in the test cases of GSM 11.10.

Example:
```
BS_MSG3_AWAIT_BEGIN ( 1, a_message, "try to receive a message" )
BS_MSG3_AWAIT_END()
```

See also: BS_MSG3_AWAIT
BS_MSG3_AWAIT_END

3.8 BS_MSG3_AWAIT_END

Description: The macro BS_MSG3_AWAIT_END marks the of the definition of an AWAIT operation. The macro BS_MSG3_AWAIT_END can only be used in conjunction with BS_MSG3_AWAIT_BEGIN.

Category: Send/Await Layer 3 messages

Syntax: BS_MSG3_AWAIT_END()

This macro has no parameters.

Example:
```
BS_MSG3_AWAIT_BEGIN ( 9, setup, SILENT )
          BF_SET_VAL ( type_of_number, NATIONAL_NUMBER, SILENT )
BS_MSG3_AWAIT_END()
```

See also: BS_MSG3_AWAIT
BS_MSG3_AWAIT_BEGIN

3.9 BS_MSG3_SEND

Description: The macro BS_MSG3_SEND is used to send a Layer 3 message at the ISS/IUT interface.
As a side effect of the SEND operation, the message is outputted on the computer screen and in a data file.

Category: Send/Await Layer 3 messages

Syntax: BS_MSG3_SEND (BS_IDX, MSG3, COM)

BS_IDX:	Index to a base station within the infrastructure simulator.
MSG3:	Layer 3 message to be sent.
COM:	A comment describing details about the SEND operation. The comment must be specified in quotation marks. There is no restriction on the length of the comment. The special comment SILENT may be used. As coding convention the first part of the comment consists of a number followed by a colon. The number is related to the "step" number of the "expected sequence" in the test cases of GSM 11.10.

Example: BS_MSG3_SEND (0, modify, SILENT)

See also: BS_MSG3_AWAIT
BS_MSG3_SEND_BEGIN
BS_MSG3_SEND_END

3.10 BS_MSG3_SEND_BEGIN

Description: The macro BS_MSG3_SEND_BEGIN is used to start the definition of a SEND operation. The macro BS_MSG3_SEND_BEGIN can only be used in conjunction with BS_MSG3_SEND_END.

Category: Send/Await Layer 3 messages

Syntax: BS_MSG3_SEND_BEGIN (BS_IDX, MSG3, COM)

BS_IDX:	Index to a base station within the infrastructure simulator.
MSG3:	Layer 3 message to be sent.
COM:	A comment describing details of the SEND operation. The comment must be specified in quotation marks. There is no restriction on the length of the comment. The special comment SILENT may be used. As coding convention the first part of the comment consists of a number followed by a colon. The number is related to the "step" number of the "expected sequence" in the test cases of GSM 11.10.

Example:

```
BS_MSG3_SEND_BEGIN ( 0, location_update_request, SILENT )
        BF_SET_VAL ( mob_id.type, M3(1,0,0), "use a TMSI" )
BS_MSG3_SEND_END()
```

See also: BS_MSG3_SEND
BS_MSG3_SEND_END

3.11 BS_MSG3_SEND_END

Description: The macro BS_MSG3_SEND_END marks the end of the definition of a SEND operation. The macro BS_MSG3_SEND_END can only be used in conjunction with BS_MSG3_SEND_BEGIN.

Category: Send/Await Layer 3 messages

Syntax: BS_MSG3_SNED_END()

This macro has no parameters.

Example: BS_MSG3_SEND_BEGIN (0, setup, SILENT)
BS_MSG3_SEND_END()

See also: BS_MSG3_SEND
BS_MSG3_SEND_BEGIN

3.12 BS_ON_OFF

Description: The macro BS_ON_OFF is used to switch a base station (BS) on or off within the infrastructure simulator (ISS).

Category: Base Station (BS) Configuration

Syntax: BS_ON_OFF (BS_IDX, ON_OFF)

BS_IDX:	Index to a base station within the infrastructure simulator.
ON_OFF:	If 0 then the BS is switched off, otherwise the BS is switched on

Example: BS_ON_OFF (0, TRUE)
The base station with index 0 is switched on.

3.13 BS_RACH_AWAIT

Description: The macro BS_RACH_AWAIT is used to receive a Layer 3 message at the ISS/IUT interface on the RACH channel (channel request message). This macro is very similar to BS_MSG3_AWAIT.

Category: Send/Await Layer 3 messages

Syntax: BS_RACH_AWAIT (BS_IDX, RACH, COM)

BS_IDX:	Index to a base station within the infrastructure simulator.
RACH:	Expected channel request message.
COM:	A comment describing details of the AWAIT operation. The comment must be specified in quotation marks. There is no restriction on the length of the comment. The special comment SILENT may be used. As coding convention the first part of the comment consists of a number followed by a colon. The number is related to the "step" number of the "expected sequence" in the test cases of GSM 11.10.

Example: BS_RACH_AWAIT (0, chan_req, SILENT)

See also: BS_MSG3_AWAIT
BS_MSG3_AWAIT_COPY
BS_RACH_AWAIT_BEGIN
BS_RACH_AWAIT_END

3.14 BS_RACH_AWAIT_COPY

Description: The macro BS_RACH_AWAIT_COPY is used to receive a Layer 3 message at the ISS/IUT interface on the RACH channel (channel request message) and to save the received message in a MSG3_VAR object. This macro is very similar to BS_RACH_AWAIT.

Category: Send/Await Layer 3 messages

Syntax: BS_RACH_AWAIT_COPY (BS_IDX, RACH, RACH_COPY, COM)

BS_IDX:	Index to a base station within the infrastructure simulator.
RACH:	Expected channel request message.
RACH_COPY:	The copy the received message.
COM:	A comment describing details of the AWAIT operation. The comment must be specified in quotation marks. There is no restriction on the length of the comment. The special comment SILENT may be used. As coding convention the first part of the comment consists of a number followed by a colon. The number is related to the "step" number of the "expected sequence" in the test cases of GSM 11.10.

Example: BS_RACH_AWAIT (0, chan_req, SILENT)

See also: BS_RACH_AWAIT

3.15 BS_RACH_AWAIT_BEGIN

Description: The macro BS_RACH_AWAIT_BEGIN is used to start the definition of an AWAIT operation. The macro BS_RACH_AWAIT_BEGIN can only be used in conjunction with BS_RACH_AWAIT_END.

Category: Send/Await Layer 3 messages

Syntax: BS_RACH_AWAIT_BEGIN (BS_IDX, RACH, COM)

BS_IDX:	Index to a base station within the infrastructure simulator.
RACH:	Expected channel request message.
COM:	A comment describing details of the AWAIT operation. The comment must be specified in quotation marks. There is no restriction on the length of the comment. The special comment SILENT may be used. As coding convention the first part of the comment consists of a number followed by a colon. The number is related to the "step" number of the "expected sequence" in the test cases of GSM 11.10.

Example:
```
BS_MSG3_AWAIT_BEGIN ( 0, the_rach_message, SILENT )
BS_MSG3_AWAIT_END()
```

See also: BS_RACH_AWAIT
BS_RACH_AWAIT_END

3.16 BS_RACH_AWAIT_END

Description: The macro BS_RACH_AWAIT_END marks the end of the definition of an AWAIT operation. The macro BS_RACH_AWAIT_END can only be used in conjunction with BS_RACH_AWAIT_BEGIN.

Category: Send/Await Layer 3 messages

Syntax: BS_RACH_AWAIT_END()

This macro has no parameters.

Example:
```
BS_RACH_AWAIT_BEGIN ( 1, rach, SILENT )
          BF_SET_VAL ( estab_cause, 0, SILENT )
BS_RACH_AWAIT_END()
```

See also: BS_RACH_AWAIT
BS_RACH_AWAIT_BEGIN

3.17 BS_RESET_SYS_INFO

Description: The macro BS_RESET_SYS_INFO is used to disable a system information message for a base station (BS) within the infrastructure simulator (ISS). After this macro call has been issued, the ISS does not send the specified system information message to the IUT.

Category: base station (BS) configuration

Syntax: BS_RESET_SYS_INFO (BS_IDX, SI)

BS_IDX:	Index to a base station within the infrastructure simulator.
SI:	A system information message. The contents of the message is ignored with the exception of the message header which contains the type of message that should be disabled.

Example: BS_RESET_SYS_INFO (2, system_information_type_1_message_type)
System Info 1 is disabled on base station 2.

See also: BS_SET_SYS_INFO

3.18 BS_SET_ARFCN

Description: The macro BS_SET_ARFCN is used to define the Absolute Radio Frequency Channel Number of a base station (BS) within the infrastructure simulator (ISS).
There is no macro to retrieve the ARFCN from the ISS.

Category: Base Station (BS) Configuration

Syntax: BS_SET_ARFCN (BS_IDX, ARFCN)

BS_IDX:	Index to a base station within the infrastructure simulator.
ARFCN:	The Absolute Radio Frequency Channel Number. In GSM 900 this is 1 <= ARFCN <= 124.

Example: BS_SET_ARFCN (0, 7)
Channel 7 is used on base station 0 (890MHz + 7*0.2MHz = 891.4 MHz, uplink GSM 900).

3.19 BS_SET_ERROR

Description: The macro BS_SET_ERROR is used to introduce a specified number of bad frames in the transmission from the ISS to the IUT.

Category: Base Station (BS) Configuration

Syntax: BS_SET_ERROR (BS_IDX, COUNT)

BS_IDX:	Index to a base station within the infrastructure simulator.
COUNT:	Number of bad frames to introduce.

Example: BS_SET_ERROR (1, 25)
25 bad frames are transmitted by base station 1 to the IUT. Then normal transmission re-sumes.

3.20 BS_SET_POWER

Description: The macro BS_SET_POWER is used to define the RF signal level received by the IUT for a base station within the Infrastructure Simulation.

Category: Base Station (BS) Configuration

Syntax: BS_SET_POWER (BS_IDX, POWER)

BS_IDX:	Index to a base station within the infrastructure simulator.
POWER:	RF signal level in dBm (the). Typical values -120 .. -30 dBm.

Example: BS_SET_POWER (0, -50)
The IUT receives -50 dBm RF level from base station 0.

3.21 BS_SET_SCH

Description: The macro BS_SET_SCH is used to define the Base Station Identification Code and the Reduced Frame Number for a base station within the infrastructure simulation. Both items are transmitted on the synchronization channel (SCH) of a base station.

Category: Base Station (BS) Configuration

Syntax: BS_SET_SCH (BS_IDX, BSIC, RFN)

BS_IDX:	Index to a base station within the infrastructure simulator.
BSIC:	The Base Station Identification Code.
RFN:	The Reduced Frame Number

Example: BS_SET_POWER (0, -50)
The IUT receives -50 dBm RF level from base station 0.

3.22 BS_SET_SYS_INFO

Description: The macro BS_SET_SYS_INFO is used to define a system information message for a base station (BS) within the infrastructure simulator (ISS).

Category: Base Station (BS) Configuration

Syntax: BS_SET_SYS_INFO (BS_IDX, SI)

BS_IDX:	Index to a base station within the infrastructure simulator.
SI:	A system information message.

Example: BS_SET_SYS_INFO (0, system_information_type_2bis_message_type)
System Info 2bis is defined on base station 0.

See also: BS_RESET_SYS_INFO

3.23 BS_STORE_RACH_PARAMS

Description: The macro BS_STORE_RACH_PARAMS is used to save the channel request message transmitted on the RACH of a base station within the infrastructure simulation.

Category: Miscellaneous

Syntax: BS_STORE_RACH_PARAMS (BS_IDX, POS)

BS_IDX:	Index to a base station within the infrastructure simulator.
POS:	Unclear, not used in the ISS implementation (VK, 27-Nov-1997).

Example: BS_STORE_RACH_PARAMS (0, 0)
The channel request message received at base station 0 is saved.

3.24 IE

Description: The IE macro is used to reference an information element during the definition of a Layer 3 message. The macro IE may only appear between MSG3_BEGIN and MSG3_END.

Category: Data Definition

Syntax: IE (IE_NAME)

IE_NAME: The name of the information element.

Examples:
```
MSG3_BEGIN ( a_sample_msg )
    IE ( an_info_element )
MSG3_END ( a_sample_msg )
```

See also: MSG3_BEGIN, MSG3_END

3.25 IE_BEGIN

Description: The IE_BEGIN macro is used to start the definition of an information element. The macro IE_BEGIN can only be used in conjunction with the macro IE_END.

Category: Data Definition

Syntax: IE_BEGIN (NAME)

NAME: The name of the information element.

Example:

```
IE_BEGIN ( a_sample_ie )
    BF ( 10, 0x3FF, ACT_NOP, ANONYMOUS, SILENT )
IE_END ( a_sample_ie )
```

See also: IE_END, MSG3_BEGIN, MSG3_END

3.26 IE_END

Description: The IE_END macro is used to mark the end of the definition of an information element. The macro IE_END can only be used in conjunction with the macro IE_BEGIN.

Category: Data Definition

Syntax: IE_END (NAME)

NAME: The name of the information element.

Example:

```
IE_BEGIN ( a_sample_ie )
    BF ( 10, 0x3FF, ACT_NOP, ANONYMOUS, SILENT )
IE_END ( a_sample_ie )
```

See also: IE_BEGIN, MSG3_BEGIN, MSG3_END

3.27 ISS_DEINIT

Description: The IE_DEINIT is used to de-initialize the infrastructure simulator. The macro IE_DEINIT should be the last TCSL macro of a test case.

Category: Infrastructure Simulator

Syntax: ISS_DEINIT()

This macro has no parameters.

Examples: ISS_DEINIT()

See also: ISS_INIT

3.28 ISS_DELAY

Description: The ISS_DELAY is used to make the ISS wait for a specified time without receiving or sending Layer 3 messages. However, certain messages such as System Information messages are still sent by the ISS.

Category: Infrastructure Simulator

Syntax: ISS_DELAY (MILLI_SEC)

MILLI_SEC: Delay in milliseconds.

Example: ISS_DELAY (120000)
Wait 2 minutes.

3.29 ISS_INIT

Description: The IE_INIT is used to initialize the infrastructure simulator (ISS). The macro IE_INIT should be the first TCSL macro of a test case.

Category: Infrastructure Simulator

Syntax: ISS_INIT (NUM_BS)

NUM_BS: Number of base stations managed by the infrastructure simulator.

Example: ISS_INIT (5)
Set up the ISS with 5 base stations.

See also: ISS_DEINIT

3.30 M1 through M8

Description: The macros M1, M2, M3, ... M8 are used to construct constants from binary bit patterns.

Category: Helper Macros

Syntax:

```
M1(B0)
M2(B1,B0)
M3(B2,B1,B0)
M4(B3,B2,B1,B0)
M5(B4,B3,B2,B1,B0)
M6(B5,B4,B3,B2,B1,B0)
M7(B6,B5,B4,B3,B2,B1,B0)
M8(B7,B6,B5,B4,B3,B2,B1,B0)
```

B0:	Bit 0 of the constant being constructed. Either "0" or "1".
B1:	Bit 1 of the constant being constructed. Either "0" or "1".
...	
B7:	Bit 7 of the constant being constructed. Either "0" or "1".

Examples: M4(1,0,1,0)
This yields the decimal number 10.

M6(1,1,1,0,1,0)
This yields the decimal hexadecimal number 0x3A.

3.31 MSG3_BEGIN

Description: The MSG3_BEGIN macro is used to start the definition of a Layer 3 message. The macro MSG3_BEGIN can only be used in conjunction with the macro MSG3_END.

Category: Data Definition

Syntax: MSG3_BEGIN (NAME)

NAME: The name of the Layer 3 message.

Example:

```
MSG3_BEGIN ( a_sample_msg3 )
   IE ( an_info_element )
MSG3_END ( a_sample_msg3 )
```

See also: IE, MSG3_END

3.32 MSG3_BF_GET_ACT

Description: The MSG3_BF_GET_ACT macro is used to retrieve the value of the "action" attribute of a bit field within a Layer 3 message.

Category: Layer 3 message manipulation

Syntax: MSG3_BF_GET_ACT (MSG3, BF_NAM)

MSG3: The name of a Layer 3 message.
BF_NAM: The name of a bit field within the Layer 3 message.

Example: MSG3_BF_GET_ACT (msg3_cpy, ie123.bf_1)

See also: MSG3_BF_SET_ACT

3.33 MSG3_BF_GET_VAL

Description: The MSG3_BF_GET_VAL macro is used to retrieve the value of a bit field within a Layer 3 message.

Category: Layer 3 message manipulation

Syntax: MSG3_BF_GET_VAL (MSG3, BF_NAM)

MSG3: The name of a Layer 3 message.
BF_NAM: The name of a bit field within the Layer 3 message.

Example: MSG3_BF_GET_VAL (rach, random_part)

See also: MSG3_BF_SET_VAL

3.34 MSG3_BF_SET_ACT

Description: The MSG3_BF_SET_ACT macro is used to define the value of the "action" attribute of a bit field within a Layer 3 message.

Category: Layer 3 message manipulation

Syntax: MSG3_BF_SET_ACT (MSG3, BF_NAM, BF_ACT)

MSG3: The name of a Layer 3 message.
The message must have been created by MSG3_COPY_CREATE.
BF_NAM: The name of a bit field within the Layer 3 message.
BF_ACT: Value of the "action" attribute. For details, refer to the description of the macro BF.

Example: MSG3_BF_SET_ACT (msg3_cpy, ie123.bf_1, ACT_FILL)
This disables the bit field bf_1 in the information element ie123 of the Layer 3 message msg3_cpy.

See also: MSG3_BF_GET_ACT

3.35 MSG3_BF_SET_VAL

Description: The MSG3_BF_SET_VAL macro is used to define the value of a bit field within a Layer 3 message.

Category: Layer 3 message manipulation

Syntax: MSG3_BF_SET_VAL (MSG3, BF_NAM, BF_VAL)

MSG3: The name of a Layer 3 message.
The message must have been created by MSG3_COPY_CREATE.
BF_NAM: The name of a bit field within the Layer 3 message.

Example: MSG3_BF_SET_VAL (rach, random_part, 0x5)

See also: MSG3_BF_GET_VAL

3.36 MSG3_COPY_CREATE

Description: The MSG3_COPY_CREATE macro is used to copy the contents of a Layer 3 message to another Layer 3 message. The destination message must have been defined with MSG3_VAR. A message that has been created with MSG3_COPY_CREATE should be released with the macro MSG3_COPY_DESTROY.

Category: Layer 3 message manipulation

Syntax: MSG3_COPY_CREATE (MSG3_COPY, MSG3)

MSG3_COPY: The name of a Layer 3 message defined by MSG3_VAR (destination).
MSG3: The name of a Layer 3 message (source).

Example: MSG3_COPY_CREATE (a_new_message, setup_message)

See also: MSG3_VAR, MSG3_COPY_DESTROY

3.37 MSG3_COPY_DESTROY

Description: The MSG3_COPY_DESTROY macro is used to destroy a Layer 3 message previously initialized with MSG3_COPY_CREATE.

Category: Layer 3 message manipulation

Syntax: MSG3_COPY_DESTROY (MSG3_COPY)

MSG3_COPY: The name of a Layer 3 message.

Example: MSG3_COPY_DESTROY (a_new_message)

See also: MSG3_VAR, MSG3_COPY_CREATE

3.38 MSG3_END

Description: The MSG3_END macro is used to mark the end of the definition of a Layer 3 message. The macro MSG3_END can only be used in conjunction with the macro MSG3_BEGIN.

Category: Layer 3 message manipulation

Syntax: MSG3_END (NAME)

NAME: The name of a Layer 3 message.

Example: refer to MSG3_BEGIN

See also: MSG3_COPY_BEGIN

3.39 MSG3_VAR

Description: The MSG3_VAR macro is used to define an empty (un-initialized) Layer 3 message.

Category: Layer 3 message manipulation

Syntax: MSG3_VAR (NAME)

NAME: The name of a Layer 3 message.

Example:

```
MSG3_BEGIN ( msg_clone )
MSG3_COPY_CREATE ( msg_clone, setup_message )
....
MSG3_COPY_DESTROY ( msg_clone )
```

See also: MSG3_BEGIN, MSG3_END

3.40 NOT_IMPLEMENTED

Description: The NOT_IMPLEMENTED macro is used to signal an item of the test specification that could not be translated into TCSL.

Category: Miscellaneous

Syntax: NOT_IMPLEMENTED (COMMENT)

COMMENT: Text that describes the feature as not implemented.
The text must be placed in quotation marks.
As coding convention, the first part of the comment consists of a number followed by a colon. The number is related to the "step" number of the "expected sequence" in the test cases of GSM 11.10.

Example: NOT_IMPLEMENTED ("12: Check the time difference. It should be less than 5 seconds\n")

3.41 SET_TIMEOUT

Description: The SET_TIMEOUT macro is used to set the maximum time the ISS should wait for a Layer 3 message from the IUT.

Category: Miscellaneous

Syntax: SET_TIMEOUT (MILLI_SEC)

MILLI_SEC: Maximum amount of time the ISS should wait for a Layer 3 message. If the time passes by without a message received, the test result is assigned the value "not passed".

Example: SET_TIMEOUT (1000)
Wait one second.

3.42 DELAY

Description: The macro DELAY is used to suspend the execution of the test program. This is useful before a send is performed to delay the send operation.

Category: Miscellaneous

Syntax: DELAY (MILLI_SEC)

MILLI_SEC: Time given in milliseconds for which the next operation is delayed.

Example: DELAY (1000)
The next operation is delayed for one second.

3.43 USER_INFO_BOX

Description: With USER_INFO_BOX an information window is created on the screen. The window shows the submitted announcement and an OK-Button. The test program will not continue before the button is clicked. This is needed when user interaction is necessary, e.g. if first the mobile must be switched on before the test may continue.

Category: Miscellaneous

Syntax: USER_INFO_BOX (ANNOUNCEMENT)

ANNOUNCEMENT: Text printed in the information window.

Example: USER_INFO_BOX (“Switch on the mobile”)

4 TCSL example

4.1 File tcsl_gsm.h : RR, MM and CC GSM messages types

```
/*-----------------------------------------------------------------*\
| Define GSM message types of RR, MM, and CC, GSM 04.08, 10.4
\*-----------------------------------------------------------------*/

#define  ADDITIONAL_ASSIGNMENT              MAKE_BYTE(0,0,1,1,1,0,1,1)
#define  IMMEDIATE_ASSIGNMENT               MAKE_BYTE(0,0,1,1,1,1,1,1)
#define  IMMEDIATE_ASSIGNMENT_EXTENDED      MAKE_BYTE(0,0,1,1,1,0,0,1)
#define  IMMEDIATE_ASSIGNMENT_REJECT        MAKE_BYTE(0,0,1,1,1,0,1,0)

#define  CIPHERING_MODE_COMMAND             MAKE_BYTE(0,0,1,1,0,1,0,1)
#define  CIPHERING_MODE_COMPLETE            MAKE_BYTE(0,0,1,1,0,0,1,0)

#define  ASSIGNMENT_COMMAND                 MAKE_BYTE(0,0,1,0,1,1,1,0)
#define  ASSIGNMENT_COMPLETE                MAKE_BYTE(0,0,1,0,1,0,0,1)
#define  ASSIGNMENT_FAILURE                 MAKE_BYTE(0,0,1,0,1,1,1,1)
#define  HANDOVER_COMMAND                   MAKE_BYTE(0,0,1,0,1,0,1,1)
#define  HANDOVER_COMPLETE                  MAKE_BYTE(0,0,1,0,1,1,0,0)
#define  HANDOVER_FAILURE                   MAKE_BYTE(0,0,1,0,1,0,0,0)
#define  PHYSICAL_INFORMATION               MAKE_BYTE(0,0,1,0,1,1,0,1)

#define  CHANNEL_RELEASE                    MAKE_BYTE(0,0,0,0,1,1,0,1)
#define  PARTIAL_RELEASE                    MAKE_BYTE(0,0,0,0,1,0,1,0)
#define  PARTIAL_RELEASE_COMPLETE           MAKE_BYTE(0,0,0,0,1,1,1,1)

#define  PAGING_REQUEST_TYPE_1              MAKE_BYTE(0,0,1,0,0,0,0,1)
#define  PAGING_REQUEST_TYPE_2              MAKE_BYTE(0,0,1,0,0,0,1,0)
#define  PAGING_REQUEST_TYPE_3              MAKE_BYTE(0,0,1,0,0,1,0,0)
#define  PAGING_RESPONSE                    MAKE_BYTE(0,0,1,0,0,1,1,1)
#define  NOTIFICATION_NCH_TYPE_1            MAKE_BYTE(0,0,1,0,0,0,0,0)
#define  NOTIFICATION_NCH_TYPE_2            MAKE_BYTE(0,0,1,0,0,0,1,1)
#define  NOTIFICATION_FACCH                 MAKE_BYTE(0,0,1,0,0,1,0,1)
#define  NOTIFICATION_SACCH                 MAKE_BYTE(0,0,1,0,0,1,1,0)
#define  NOTIFICATION_RESPONSE              MAKE_BYTE(0,0,0,0,1,0,1,1)

#define  SYSTEM_INFORMATION_TYPE_8          MAKE_BYTE(0,0,0,1,1,0,0,0)
#define  SYSTEM_INFORMATION_TYPE_1          MAKE_BYTE(0,0,0,1,1,0,0,1)
#define  SYSTEM_INFORMATION_TYPE_2          MAKE_BYTE(0,0,0,1,1,0,1,0)
#define  SYSTEM_INFORMATION_TYPE_3          MAKE_BYTE(0,0,0,1,1,0,1,1)
#define  SYSTEM_INFORMATION_TYPE_4          MAKE_BYTE(0,0,0,1,1,1,0,0)
#define  SYSTEM_INFORMATION_TYPE_5          MAKE_BYTE(0,0,0,1,1,1,0,1)
#define  SYSTEM_INFORMATION_TYPE_6          MAKE_BYTE(0,0,0,1,1,1,1,0)
#define  SYSTEM_INFORMATION_TYPE_7          MAKE_BYTE(0,0,0,1,1,1,1,1)

#define  SYSTEM_INFORMATION_TYPE_2bis       MAKE_BYTE(0,0,0,0,0,0,1,0)
#define  SYSTEM_INFORMATION_TYPE_2ter       MAKE_BYTE(0,0,0,0,0,0,1,1)
#define  SYSTEM_INFORMATION_TYPE_5bis       MAKE_BYTE(0,0,0,0,0,1,0,1)
#define  SYSTEM_INFORMATION_TYPE_5ter       MAKE_BYTE(0,0,0,0,0,1,1,0)
#define  SYSTEM_INFORMATION_TYPE_9          MAKE_BYTE(0,0,0,0,0,1,0,0)
#define  SYSTEM_INFORMATION_TYPE_10         MAKE_BYTE(0,0,0,0,0,0,0,0)
#define  SYSTEM_INFORMATION_TYPE_10bis      MAKE_BYTE(0,0,0,0,0,0,0,1)
#define  SYSTEM_INFORMATION_TYPE_11         MAKE_BYTE(0,0,0,0,0,1,1,1)
#define  SYSTEM_INFORMATION_TYPE_12         MAKE_BYTE(0,0,0,0,1,0,0,0)

#define  CHANNEL_MODE_MODIFY                MAKE_BYTE(0,0,0,1,0,0,0,0)
#define  RR_STATUS                          MAKE_BYTE(0,0,0,1,0,0,1,0)
#define  CHANNEL_MODE_MODIFY_ACKNOWLEDGE    MAKE_BYTE(0,0,0,1,0,1,1,1)
#define  FREQUENCY_REDEFINITION             MAKE_BYTE(0,0,0,1,0,1,0,0)
#define  MEASUREMENT_REPORT                 MAKE_BYTE(0,0,0,1,0,1,0,1)
#define  CLASSMARK_CHANGE                   MAKE_BYTE(0,0,0,1,0,1,1,0)
#define  CLASSMARK_ENQUIRY                  MAKE_BYTE(0,0,0,1,0,0,1,1)

#define  VGCS_UPLINK_GRANT                  MAKE_BYTE(0,0,0,0,1,0,0,1)
#define  UPLINK_RELEASE                     MAKE_BYTE(0,0,0,0,1,1,1,0)
#define  UPLINK_FREE                        MAKE_BYTE(0,0,0,0,1,1,0,0)
#define  UPLINK_BUSY                        MAKE_BYTE(0,0,1,0,1,0,1,0)
#define  TALKER_INDICATION                  MAKE_BYTE(0,0,0,1,0,0,0,1)

#define  IMSI_DETACH_INDICATION             MAKE_BYTE(0,0,0,0,0,0,0,1)
#define  LOCATION_UPDATING_ACCEPT           MAKE_BYTE(0,0,0,0,0,0,1,0)
#define  LOCATION_UPDATING_REJECT           MAKE_BYTE(0,0,0,0,0,1,0,0)
#define  LOCATION_UPDATING_REQUEST          MAKE_BYTE(0,0,0,0,1,0,0,0)

#define  AUTHENTICATION_REJECT              MAKE_BYTE(0,0,0,1,0,0,0,1)
#define  AUTHENTICATION_REQUEST             MAKE_BYTE(0,0,0,1,0,0,1,0)
#define  AUTHENTICATION_RESPONSE            MAKE_BYTE(0,0,0,1,0,1,0,0)
#define  IDENTITY_REQUEST                   MAKE_BYTE(0,0,0,1,1,0,0,0)
#define  IDENTITY_RESPONSE                  MAKE_BYTE(0,0,0,1,1,0,0,1)
#define  TMSI_REALLOCATION_COMMAND          MAKE_BYTE(0,0,0,1,1,0,1,0)
#define  TMSI_REALLOCATION_COMPLETE         MAKE_BYTE(0,0,0,1,1,0,1,1)

#define  CM_SERVICE_ACCEPT                  MAKE_BYTE(0,0,1,0,0,0,0,1)
#define  CM_SERVICE_REJECT                  MAKE_BYTE(0,0,1,0,0,0,1,0)
#define  CM_SERVICE_ABORT                   MAKE_BYTE(0,0,1,0,0,0,1,1)
#define  CM_SERVICE_REQUEST                 MAKE_BYTE(0,0,1,0,0,1,0,0)
#define  CM_REESTABLISHMENT_REQUEST         MAKE_BYTE(0,0,1,0,1,0,0,0)
#define  ABORT                              MAKE_BYTE(0,0,1,0,1,0,0,1)

#define  MM_NULL                            MAKE_BYTE(0,0,1,1,0,0,0,0)
#define  MM_STATUS                          MAKE_BYTE(0,0,1,1,0,0,0,1)
#define  MM_INFORMATION                     MAKE_BYTE(0,0,1,1,0,0,1,0)

#define  ESCAPE_NATIONAL_SPEC_MSG_TYPE      MAKE_BYTE(0,0,0,0,0,0,0,0)

#define  ALERTING                           MAKE_BYTE(0,0,0,0,0,0,0,1)
#define  CALL_CONFIRMED                     MAKE_BYTE(0,0,0,0,1,0,0,0)
#define  CALL_PROCEEDING                    MAKE_BYTE(0,0,0,0,0,0,1,0)
```

```
#define  CONNECT                           MAKE_BYTE(0,0,0,0,0,1,1,1)
#define  CONNECT_ACKNOWLEDGE               MAKE_BYTE(0,0,0,0,1,1,1,1)
#define  EMERGENCY_SETUP                   MAKE_BYTE(0,0,0,0,1,1,1,0)
#define  PROGRESS                          MAKE_BYTE(0,0,0,0,0,0,1,1)
#define  SETUP                             MAKE_BYTE(0,0,0,0,0,1,0,1)

#define  MODIFY                            MAKE_BYTE(0,0,0,1,0,1,1,1)
#define  MODIFY_COMPLETE                   MAKE_BYTE(0,0,0,1,1,1,1,1)
#define  MODIFY_REJECT                     MAKE_BYTE(0,0,0,1,0,0,1,1)
#define  USER_INFORMATION                  MAKE_BYTE(0,0,0,1,0,0,0,0)
#define  HOLD                              MAKE_BYTE(0,0,0,1,1,0,0,0)
#define  HOLD_ACKNOWLEDGE                  MAKE_BYTE(0,0,0,1,1,0,0,1)
#define  HOLD_REJECT                       MAKE_BYTE(0,0,0,1,1,0,1,0)
#define  RETRIEVE                          MAKE_BYTE(0,0,0,1,1,1,0,0)
#define  RETRIEVE_ACKNOWLEDGE              MAKE_BYTE(0,0,0,1,1,1,0,1)
#define  RETRIEVE_REJECT                   MAKE_BYTE(0,0,0,1,1,1,1,0)

#define  DISCONNECT                        MAKE_BYTE(0,0,1,0,0,1,0,1)
#define  RELEASE                           MAKE_BYTE(0,0,1,0,1,1,0,1)
#define  RELEASE_COMPLETE                  MAKE_BYTE(0,0,1,0,1,0,1,0)

#define  CONGESTION_CONTROL                MAKE_BYTE(0,0,1,1,1,0,0,1)
#define  NOTIFY                            MAKE_BYTE(0,0,1,1,1,1,1,0)
#define  STATUS                            MAKE_BYTE(0,0,1,1,1,1,0,1)
#define  STATUS_ENQUIRY                    MAKE_BYTE(0,0,1,1,0,1,0,0)
#define  START_DTMF                        MAKE_BYTE(0,0,1,1,0,1,0,1)
#define  STOP_DTMF                         MAKE_BYTE(0,0,1,1,0,0,0,1)
#define  STOP_DTMF_ACKNOWLEDGE             MAKE_BYTE(0,0,1,1,0,0,1,0)
#define  START_DTMF_ACKNOWLEDGE            MAKE_BYTE(0,0,1,1,0,1,1,0)
#define  START_DTMF_REJECT                 MAKE_BYTE(0,0,1,1,0,1,1,1)
#define  FACILITY                          MAKE_BYTE(0,0,1,1,1,0,1,0)

/*------------------------------------------------------------------*\
| Common constants
\*------------------------------------------------------------------*/

   #define  REST_OCTET                     MAKE_BYTE(0,0,1,0,1,0,1,1)

/*------------------------------------------------------------------*\
| Common Information Elements
\*------------------------------------------------------------------*/

   /*---------------------------------------------------------------*\
   | Protocol Discriminator for RR, MM, CC (GSM 04.08, 10.2)
   \*---------------------------------------------------------------*/
   IE_BEGIN(call_control_protocol_discriminator)
      BF(4,M4(0,0,1,1),ACT_NOP,"PD","CC protocol discriminator")
   IE_END(call_control_protocol_discriminator)

   IE_BEGIN(mobility_management_protocol_discriminator)
      BF(4,M4(0,1,0,1),ACT_NOP,"PD","MM protocol discriminator")
   IE_END(mobility_management_protocol_discriminator)

   IE_BEGIN(rr_management_protocol_discriminator)
      BF(4,M4(0,1,1,0),ACT_NOP,"PD","RR protocol discriminator")
   IE_END(rr_management_protocol_discriminator)

   IE_BEGIN(skip_indicator)
      BF(4,0,ACT_NOP,ANONYM,"skip indicator")
   IE_END(skip_indicator)

   IE_BEGIN(spare_half_octet)
      BF(4,0,ACT_NOP,ANONYM,"spare half octet")
   IE_END(spare_half_octet)

/*------------------------------------------------------------------*\
| Generate all Information Elements containing Message Types
| The Message Types have "ACT_CHECK" assigned.
\*------------------------------------------------------------------*/

#define MAKE_MESSAGE_TYPE_IE(COMMENT,IE_VAL,IE_NAM)\
   IE_BEGIN(IE_NAM)\
      BF(8,IE_VAL,ACT_CHECK,ANONYM,COMMENT)\
   IE_END(IE_NAM)

MAKE_MESSAGE_TYPE_IE("RR",ADDITIONAL_ASSIGNMENT          ,additional_assignment_message_type          )
MAKE_MESSAGE_TYPE_IE("RR",IMMEDIATE_ASSIGNMENT           ,immediate_assignment_message_type           )
MAKE_MESSAGE_TYPE_IE("RR",IMMEDIATE_ASSIGNMENT_EXTENDED  ,immediate_assignment_extended_message_type  )
MAKE_MESSAGE_TYPE_IE("RR",IMMEDIATE_ASSIGNMENT_REJECT    ,immediate_assignment_reject_message_type    )

MAKE_MESSAGE_TYPE_IE("RR",CIPHERING_MODE_COMMAND         ,ciphering_mode_command_message_type         )
MAKE_MESSAGE_TYPE_IE("RR",CIPHERING_MODE_COMPLETE        ,ciphering_mode_complete_message_type        )

MAKE_MESSAGE_TYPE_IE("RR",ASSIGNMENT_COMMAND             ,assignment_command_message_type             )
MAKE_MESSAGE_TYPE_IE("RR",ASSIGNMENT_COMPLETE            ,assignment_complete_message_type            )
MAKE_MESSAGE_TYPE_IE("RR",ASSIGNMENT_FAILURE             ,assignment_failure_message_type             )
MAKE_MESSAGE_TYPE_IE("RR",HANDOVER_COMMAND               ,handover_command_message_type               )
MAKE_MESSAGE_TYPE_IE("RR",HANDOVER_COMPLETE              ,handover_complete_message_type              )
MAKE_MESSAGE_TYPE_IE("RR",HANDOVER_FAILURE               ,handover_failure_message_type               )
MAKE_MESSAGE_TYPE_IE("RR",PHYSICAL_INFORMATION           ,physical_information_message_type           )

MAKE_MESSAGE_TYPE_IE("RR",CHANNEL_RELEASE                ,channel_release_message_type                )
MAKE_MESSAGE_TYPE_IE("RR",PARTIAL_RELEASE                ,partial_release_message_type                )
MAKE_MESSAGE_TYPE_IE("RR",PARTIAL_RELEASE_COMPLETE       ,partial_release_complete_message_type       )

MAKE_MESSAGE_TYPE_IE("RR",PAGING_REQUEST_TYPE_1          ,paging_request_type_1_message_type          )
MAKE_MESSAGE_TYPE_IE("RR",PAGING_REQUEST_TYPE_2          ,paging_request_type_2_message_type          )
MAKE_MESSAGE_TYPE_IE("RR",PAGING_REQUEST_TYPE_3          ,paging_request_type_3_message_type          )
MAKE_MESSAGE_TYPE_IE("RR",PAGING_RESPONSE                ,paging_response_message_type                )
MAKE_MESSAGE_TYPE_IE("RR",NOTIFICATION_NCH_TYPE_1        ,notification_nch_type_1_message_type        )
MAKE_MESSAGE_TYPE_IE("RR",NOTIFICATION_NCH_TYPE_2        ,notification_nch_type_2_message_type        )
MAKE_MESSAGE_TYPE_IE("RR",NOTIFICATION_FACCH             ,notification_facch_message_type             )
MAKE_MESSAGE_TYPE_IE("RR",NOTIFICATION_SACCH             ,notification_sacch_message_type             )
MAKE_MESSAGE_TYPE_IE("RR",NOTIFICATION_RESPONSE          ,notification_response_message_type          )

MAKE_MESSAGE_TYPE_IE("RR",SYSTEM_INFORMATION_TYPE_8      ,system_information_type_8_message_type      )
MAKE_MESSAGE_TYPE_IE("RR",SYSTEM_INFORMATION_TYPE_1      ,system_information_type_1_message_type      )
MAKE_MESSAGE_TYPE_IE("RR",SYSTEM_INFORMATION_TYPE_2      ,system_information_type_2_message_type      )
MAKE_MESSAGE_TYPE_IE("RR",SYSTEM_INFORMATION_TYPE_3      ,system_information_type_3_message_type      )
MAKE_MESSAGE_TYPE_IE("RR",SYSTEM_INFORMATION_TYPE_4      ,system_information_type_4_message_type      )
MAKE_MESSAGE_TYPE_IE("RR",SYSTEM_INFORMATION_TYPE_5      ,system_information_type_5_message_type      )
MAKE_MESSAGE_TYPE_IE("RR",SYSTEM_INFORMATION_TYPE_6      ,system_information_type_6_message_type      )
MAKE_MESSAGE_TYPE_IE("RR",SYSTEM_INFORMATION_TYPE_7      ,system_information_type_7_message_type      )

MAKE_MESSAGE_TYPE_IE("RR",SYSTEM_INFORMATION_TYPE_2bis   ,system_information_type_2bis_message_type   )
MAKE_MESSAGE_TYPE_IE("RR",SYSTEM_INFORMATION_TYPE_2ter   ,system_information_type_2ter_message_type   )
MAKE_MESSAGE_TYPE_IE("RR",SYSTEM_INFORMATION_TYPE_5bis   ,system_information_type_5bis_message_type   )
MAKE_MESSAGE_TYPE_IE("RR",SYSTEM_INFORMATION_TYPE_5ter   ,system_information_type_5ter_message_type   )
MAKE_MESSAGE_TYPE_IE("RR",SYSTEM_INFORMATION_TYPE_9      ,system_information_type_9_message_type      )
MAKE_MESSAGE_TYPE_IE("RR",SYSTEM_INFORMATION_TYPE_10     ,system_information_type_10_message_type     )
MAKE_MESSAGE_TYPE_IE("RR",SYSTEM_INFORMATION_TYPE_10bis  ,system_information_type_10bis_message_type  )
MAKE_MESSAGE_TYPE_IE("RR",SYSTEM_INFORMATION_TYPE_11     ,system_information_type_11_message_type     )
MAKE_MESSAGE_TYPE_IE("RR",SYSTEM_INFORMATION_TYPE_12     ,system_information_type_12_message_type     )

MAKE_MESSAGE_TYPE_IE("RR",CHANNEL_MODE_MODIFY            ,channel_mode_modify_message_type            )
MAKE_MESSAGE_TYPE_IE("RR",RR_STATUS                      ,rr_status_message_type                      )
MAKE_MESSAGE_TYPE_IE("RR",CHANNEL_MODE_MODIFY_ACKNOWLEDGE,channel_mode_modify_acknowledge_message_type)
MAKE_MESSAGE_TYPE_IE("RR",FREQUENCY_REDEFINITION         ,frequency_redefinition_message_type         )
MAKE_MESSAGE_TYPE_IE("RR",MEASUREMENT_REPORT             ,measurement_report_message_type             )
MAKE_MESSAGE_TYPE_IE("RR",CLASSMARK_CHANGE               ,classmark_change_message_type               )
MAKE_MESSAGE_TYPE_IE("RR",CLASSMARK_ENQUIRY              ,classmark_enquiry_message_type              )

MAKE_MESSAGE_TYPE_IE("RR",VGCS_UPLINK_GRANT              ,vgcs_uplink_grant_message_type              )
MAKE_MESSAGE_TYPE_IE("RR",UPLINK_RELEASE                 ,uplink_release_message_type                 )
MAKE_MESSAGE_TYPE_IE("RR",UPLINK_FREE                    ,uplink_free_message_type                    )
MAKE_MESSAGE_TYPE_IE("RR",UPLINK_BUSY                    ,uplink_busy_message_type                    )
MAKE_MESSAGE_TYPE_IE("RR",TALKER_INDICATION              ,talker_indication_message_type              )

MAKE_MESSAGE_TYPE_IE("MM",IMSI_DETACH_INDICATION         ,imsi_detach_indication_message_type         )
MAKE_MESSAGE_TYPE_IE("MM",LOCATION_UPDATING_ACCEPT       ,location_updating_accept_message_type       )
MAKE_MESSAGE_TYPE_IE("MM",LOCATION_UPDATING_REJECT       ,location_updating_reject_message_type       )
MAKE_MESSAGE_TYPE_IE("MM",LOCATION_UPDATING_REQUEST      ,location_updating_reque st_message_type      )

MAKE_MESSAGE_TYPE_IE("MM",AUTHENTICATION_REJECT          ,authentication_reject_message_type          )
MAKE_MESSAGE_TYPE_IE("MM",AUTHENTICATION_REQUEST         ,authentication_request_message_type         )
MAKE_MESSAGE_TYPE_IE("MM",AUTHENTICATION_RESPONSE        ,authentication_response_message_type        )
MAKE_MESSAGE_TYPE_IE("MM",IDENTITY_REQUEST               ,identity_request_message_type               )
MAKE_MESSAGE_TYPE_IE("MM",IDENTITY_RESPONSE              ,identity_response_message_type              )
MAKE_MESSAGE_TYPE_IE("MM",TMSI_REALLOCATION_COMMAND      ,tmsi_reallocation_command_message_type      )
MAKE_MESSAGE_TYPE_IE("MM",TMSI_REALLOCATION_COMPLETE     ,tmsi_reallocation_complete_message_type     )

MAKE_MESSAGE_TYPE_IE("MM",CM_SERVICE_ACCEPT              ,cm_service_accept_message_type              )
MAKE_MESSAGE_TYPE_IE("MM",CM_SERVICE_REJECT              ,cm_service_reject_message_type              )
MAKE_MESSAGE_TYPE_IE("MM",CM_SERVICE_ABORT               ,cm_service_abort_message_type               )
MAKE_MESSAGE_TYPE_IE("MM",CM_SERVICE_REQUEST             ,cm_service_request_message_type             )
MAKE_MESSAGE_TYPE_IE("MM",CM_REESTABLISHMENT_REQUEST     ,cm_reestablishment_request_message_type     )
MAKE_MESSAGE_TYPE_IE("MM",ABORT                          ,abort_message_type                          )

MAKE_MESSAGE_TYPE_IE("MM",MM_NULL                        ,mm_null_message_type                        )
MAKE_MESSAGE_TYPE_IE("MM",MM_STATUS                      ,mm_status_message_type                      )
MAKE_MESSAGE_TYPE_IE("MM",MM_INFORMATION                 ,mm_information_message_type                 )

MAKE_MESSAGE_TYPE_IE("CC",ESCAPE_NATIONAL_SPEC_MSG_TYPE  ,escape_national_spec_msg_type_message_type   )

MAKE_MESSAGE_TYPE_IE("CC",ALERTING                       ,alerting_message_type                       )
MAKE_MESSAGE_TYPE_IE("CC",CALL_CONFIRMED                 ,call_confirmed_message_type                 )
MAKE_MESSAGE_TYPE_IE("CC",CALL_PROCEEDING                ,call_proceeding_message_type                )
MAKE_MESSAGE_TYPE_IE("CC",CONNECT                        ,connect_message_type                        )
MAKE_MESSAGE_TYPE_IE("CC",CONNECT_ACKNOWLEDGE            ,connect_acknowledge_message_type            )
MAKE_MESSAGE_TYPE_IE("CC",EMERGENCY_SETUP                ,emergency_setup_message_type                )
MAKE_MESSAGE_TYPE_IE("CC",PROGRESS                       ,progress_message_type                       )
MAKE_MESSAGE_TYPE_IE("CC",SETUP                          ,setup_message_type                          )

MAKE_MESSAGE_TYPE_IE("CC",MODIFY                         ,modify_message_type                         )
MAKE_MESSAGE_TYPE_IE("CC",MODIFY_COMPLETE                ,modify_complete_message_type                )
MAKE_MESSAGE_TYPE_IE("CC",MODIFY_REJECT                  ,modify_reject_message_type                  )
MAKE_MESSAGE_TYPE_IE("CC",USER_INFORMATION               ,user_information_message_type               )
MAKE_MESSAGE_TYPE_IE("CC",HOLD                           ,hold_message_type                           )
MAKE_MESSAGE_TYPE_IE("CC",HOLD_ACKNOWLEDGE               ,hold_acknowledge_message_type               )
MAKE_MESSAGE_TYPE_IE("CC",HOLD_REJECT                    ,hold_reject_message_type                    )
MAKE_MESSAGE_TYPE_IE("CC",RETRIEVE                       ,retrieve_message_type                       )
MAKE_MESSAGE_TYPE_IE("CC",RETRIEVE_ACKNOWLEDGE           ,retrieve_acknowledge_message_type           )
MAKE_MESSAGE_TYPE_IE("CC",RETRIEVE_REJECT                ,retrieve_reject_message_type                )

MAKE_MESSAGE_TYPE_IE("CC",DISCONNECT                     ,disconnect_message_type                     )
MAKE_MESSAGE_TYPE_IE("CC",RELEASE                        ,release_message_type                        )
MAKE_MESSAGE_TYPE_IE("CC",RELEASE_COMPLETE               ,release_complete_message_type               )

MAKE_MESSAGE_TYPE_IE("CC",CONGESTION_CONTROL             ,congestion_control_message_type             )
MAKE_MESSAGE_TYPE_IE("CC",NOTIFY                         ,notify_message_type                         )
MAKE_MESSAGE_TYPE_IE("CC",STATUS                         ,status_message_type                         )
MAKE_MESSAGE_TYPE_IE("CC",STATUS_ENQUIRY                 ,status_enquiry_message_type                 )
MAKE_MESSAGE_TYPE_IE("CC",START_DTMF                     ,start_dtmf_message_type                     )
MAKE_MESSAGE_TYPE_IE("CC",STOP_DTMF                      ,stop_dtmf_message_type                      )
MAKE_MESSAGE_TYPE_IE("CC",STOP_DTMF_ACKNOWLEDGE          ,stop_dtmf_acknowledge_message_type          )
MAKE_MESSAGE_TYPE_IE("CC",START_DTMF_ACKNOWLEDGE         ,start_dtmf_acknowledge_message_type         )
MAKE_MESSAGE_TYPE_IE("CC",START_DTMF_REJECT              ,start_dtmf_reject_message_type              )
MAKE_MESSAGE_TYPE_IE("CC",FACILITY                       ,facility_message_type                       )
```

4.2 File 10_1_2.h : System information messages (MTC)

```
/*------------------------------------------------------------------*\
| GSM 11.10
| 10 Generic call set up procedure
| 10.1 Generic call setup-up procedure for mobile terminating speech calls
| 10.1.2 Definition of system information messages
\*------------------------------------------------------------------*/

/*------------------------------------------------------------------*\
| Information Elements
\*------------------------------------------------------------------*/

   /*--------------------------------------------------------------*\
   | BCCH Frequency list:
   | Indicates seven surrounding cells on any ARFCN of the supported
   | band, excluding ARFCNs in or immediately adjacent to those
   | specified in section 6.2 (GSM 11.10).
   | From GSM 11.10, section6.2 the following ARFCN are given :
   |     10, 14, 17, 18, 22, 24, 26, 30, 31, 34, 38, 42, 45, 46, 50,
   |     52, 54, 58, 59, 62, 66, 70, 73, 74, 78, 80, 82, 86, 87, 90,
   |     94, 98, 101, 102, 106, 108, 110, 114
   | The following 7 cells are chosen :
   |            121,117,       76,     48,     12,7,1
   | Thus BA is : 01100000  00000800 00008000 00000841
   \*--------------------------------------------------------------*/
   IE_BEGIN(bcch_frequency_list)
      BF(32,0x01100000,ACT_NOP,ANONYM,"bit 128 thru  97")
      BF(32,0x00000800,ACT_NOP,ANONYM,"bit  96 thru  65")
      BF(32,0x00008000,ACT_NOP,ANONYM,"bit  64 thru  33")
      BF(32,0x00000841,ACT_NOP,ANONYM,"bit  32 thru   1")
   IE_END(bcch_frequency_list)

   IE_BEGIN(cell_channel_description)
      BF(32,0x00000000,ACT_NOP,ANONYM,"Includes the       ")
      BF(32,0x00000000,ACT_NOP,ANONYM,"hopping sequence   ")
      BF(32,0x00000000,ACT_NOP,ANONYM,"ARFCNs, if hopping ")
      BF(32,0x00000000,ACT_NOP,ANONYM,"is used.           ")
   IE_END(cell_channel_description)

   IE_BEGIN(cell_identity)
      BF(16,0x0001,ACT_NOP,ANONYM,"CI VALUE 0001 hex (not relevant)")
   IE_END(cell_identity)

   IE_BEGIN(cell_options)
      BF(1,0,ACT_NOP,ANONYM              ,"spare               ")
      BF(1,0,ACT_NOP,pwrc                ,"power control not set")
      BF(2,2,ACT_NOP,dtx                 ,"MS must not use DTX  ")
      BF(4,1,ACT_NOP,radio_link_time_out ,"8                    ")
   IE_END(cell_options)

   IE_BEGIN(cell_selection_parameter)
      BF(3,       0,ACT_NOP,cell_reselect_hysteresis,"0 dB                        ")
      BF(5,       0,ACT_NOP,ms_txpwr_max_cch        ,"Max. output power of MS      ")
      BF(1,       0,ACT_NOP,acs                     ,"no additional cell parameters")
      BF(1,       0,ACT_NOP,neci                    ,"New establishment cause not supported")
      BF(6,-90+111,ACT_NOP,rxlev_access_min        ,"-90 dBm")
   IE_END(cell_selection_parameter)

   IE_BEGIN(control_channel_description)
      BF(1,0,ACT_NOP,ANONYM        ,"spare                            ")
      BF(1,0,ACT_NOP,att           ,"MS shall not apply (not relevant) ")
      BF(3,0,ACT_NOP,bs_ag_blks_res,"0 blocks reserved (not relevant)  ")
      BF(3,1,ACT_NOP,ccch_conf     ,"Combined CCCH/SDCCH (not relevant)")
      BF(5,0,ACT_NOP,ANONYM        ,"spare                            ")
      BF(3,3,ACT_NOP,bs_pa_mfrms   ,"5 multiframes (not relevant)      ")
      BF(8,0,ACT_NOP,t3212         ,"Infinite                         ")
   IE_END(control_channel_description)

   IE_BEGIN(l2_pseudo_length_12)
      BF(6,12,ACT_NOP,ANONYM,SILENT)
      BF(1, 0,ACT_NOP,ANONYM,SILENT)
      BF(1, 1,ACT_NOP,ANONYM,SILENT)
   IE_END(l2_pseudo_length_12)

   IE_BEGIN(l2_pseudo_length_18)
      BF(6,18,ACT_NOP,ANONYM,SILENT)
      BF(1, 0,ACT_NOP,ANONYM,SILENT)
      BF(1, 1,ACT_NOP,ANONYM,SILENT)
   IE_END(l2_pseudo_length_18)

   IE_BEGIN(l2_pseudo_length_21)
      BF(6,21,ACT_NOP,ANONYM,SILENT)
      BF(1, 0,ACT_NOP,ANONYM,SILENT)
      BF(1, 1,ACT_NOP,ANONYM,SILENT)
   IE_END(l2_pseudo_length_21)

   IE_BEGIN(l2_pseudo_length_22)
      BF(6,22,ACT_NOP,ANONYM,SILENT)
      BF(1, 0,ACT_NOP,ANONYM,SILENT)
      BF(1, 1,ACT_NOP,ANONYM,SILENT)
   IE_END(l2_pseudo_length_22)

      #define  MCC   1       /* 001  decimal (not relevant)   */
      #define  MNC   1       /* 01   decimal (not relevant)   */
      #define  LAC   0x0001   /* 0001 hex (not relevant)       */

   IE_BEGIN(location_area_identification)
      BF( 4,MCC>>4,ACT_NOP,mcc_dig_2,"digit 2 of mobile country code")
      BF( 4,MCC>>0,ACT_NOP,mcc_dig_1,"digit 1 of mobile country code")
      BF( 4,   0xF,ACT_NOP,ANONYM   ,"end of MCC                    ")
      BF( 4,MCC>>8,ACT_NOP,mcc_dig_3,"digit 3 of mobile country code")
      BF( 4,MNC>>4,ACT_NOP,mnc_dig_2,"digit 2 of mobile network code")
      BF( 4,MNC>>0,ACT_NOP,mnc_dig_1,"digit 1 of mobile network code")
      BF(16,   LAC,ACT_NOP,lac      ,"Location area code            ")
   IE_END(location_area_identification)

   IE_BEGIN(ncc_permitted)
      BF(8,0xFF,ACT_NOP,ncc_permit,"e.g. all NCCs permitted")
   IE_END(ncc_permitted)

   IE_BEGIN(rach_control_parameter)
      BF(  2,0,ACT_NOP,max_retrans                ,"Any Value  ")
      BF(  4,0,ACT_NOP,tx_integer                 ,"Any Value  ")
      BF(  1,0,ACT_NOP,cell_bar_access            ,"Not barred ")
      BF(  1,1,ACT_NOP,call_re_establishment      ,"Not Allowed")
      BF(  5,0,ACT_NOP,access_control_class_15_11 ,"None Barred")
      BF(  1,0,ACT_NOP,emergency_call             ,"Allowed    ")
      BF( 10,0,ACT_NOP,access_control_class_09_00 ,"None Barred")
   IE_END(rach_control_parameter)

   IE_BEGIN(si_1_rest_octets)
      BF(8,REST_OCTET,ACT_NOP,ANONYM,"Spare Octets")
   IE_END(si_1_rest_octets)

   IE_BEGIN(si_3_rest_octets) /* optionally contains cell (re)select parameter */
      BF(1,                0,ACT_NOP,p1    ,"C2 parameters not present")
      BF(7,REST_OCTET & 0x7F,ACT_NOP,ANONYM,SILENT                     )
      BF(8,REST_OCTET       ,ACT_NOP,ANONYM,SILENT                     )
      BF(8,REST_OCTET       ,ACT_NOP,ANONYM,SILENT                     )
      BF(8,REST_OCTET       ,ACT_NOP,ANONYM,SILENT                     )
   IE_END(si_3_rest_octets)

   IE_BEGIN(si_4_rest_octets) /* optionally contains cell (re)select parameter */
      BF(1,                0,ACT_NOP,p1    ,"C2 parameters not present")
      BF(7,REST_OCTET & 0x7F,ACT_NOP,ANONYM,SILENT                     )
      BF(8,REST_OCTET       ,ACT_NOP,ANONYM,SILENT                     )
      BF(8,REST_OCTET       ,ACT_NOP,ANONYM,SILENT                     )
      BF(8,REST_OCTET       ,ACT_NOP,ANONYM,SILENT                     )
      BF(8,REST_OCTET       ,ACT_NOP,ANONYM,SILENT                     )
      BF(8,REST_OCTET       ,ACT_NOP,ANONYM,SILENT                     )
      BF(8,REST_OCTET       ,ACT_NOP,ANONYM,SILENT                     )
      BF(8,REST_OCTET       ,ACT_NOP,ANONYM,SILENT                     )
      BF(8,REST_OCTET       ,ACT_NOP,ANONYM,SILENT                     )
      BF(8,REST_OCTET       ,ACT_NOP,ANONYM,SILENT                     )
   IE_END(si_4_rest_octets)

/*------------------------------------------------------------------*\
| Messages
\*------------------------------------------------------------------*/

MSG3_BEGIN(system_information_type_1)
   IE(l2_pseudo_length_21)
   IE(skip_indicator)
   IE(rr_management_protocol_discriminator)
   IE(system_information_type_1_message_type)
   IE(cell_channel_description)
   IE(rach_control_parameter)
   IE(si_1_rest_octets)
MSG3_END(system_information_type_1)

MSG3_BEGIN(system_information_type_2)
   IE(l2_pseudo_length_22)
   IE(skip_indicator)
   IE(rr_management_protocol_discriminator)
   IE(system_information_type_2_message_type)
   IE(bcch_frequency_list)
   IE(ncc_permitted)
   IE(rach_control_parameter)
MSG3_END(system_information_type_2)

MSG3_BEGIN(system_information_type_3)
   IE(l2_pseudo_length_18)
   IE(skip_indicator)
   IE(rr_management_protocol_discriminator)
   IE(system_information_type_3_message_type)
   IE(cell_identity)
   IE(location_area_identification)
   IE(control_channel_description)
   IE(cell_options)
   IE(cell_selection_parameter)
   IE(rach_control_parameter)
   IE(si_3_rest_octets)
MSG3_END(system_information_type_3)

MSG3_BEGIN(system_information_type_4)
   IE(l2_pseudo_length_12)
   IE(skip_indicator)
   IE(rr_management_protocol_discriminator)
   IE(system_information_type_4_message_type)
   IE(location_area_identification)
   IE(cell_selection_parameter)
   IE(rach_control_parameter)
   IE(si_4_rest_octets)
MSG3_END(system_information_type_4)
```

```
MSG3_BEGIN(system_information_type_5)
   IE(skip_indicator)
   IE(rr_management_protocol_discriminator)
   IE(system_information_type_5_message_type)
   IE(bcch_frequency_list)
MSG3_END(system_information_type_5)

MSG3_BEGIN(system_information_type_6)
   IE(skip_indicator)
   IE(rr_management_protocol_discriminator)
   IE(system_information_type_6_message_type)
   IE(cell_identity)
   IE(location_area_identification)
   IE(cell_options)
   IE(ncc_permitted)
MSG3_END(system_information_type_6)
```

4.3 File 10_1_4.h : Specific message contents (MTC)

```
/*-----------------------------------------------------------------*\
| GSM 11.10
| 10 Generic call set up procedure
| 10.1 Generic call setup-up procedure for mobile terminating speech calls
| 10.1.4 Specific message contents
\*-----------------------------------------------------------------*/

/*-----------------------------------------------------------------*\
| Information Elements
\*-----------------------------------------------------------------*/

   IE_BEGIN(authentication_parameter_rand)
      BF(32,0x80000000,ACT_NOP,rand_127_096,SILENT)
      BF(32,0x00000012,ACT_NOP,rand_095_064,SILENT)
      BF(32,0x34000000,ACT_NOP,rand_063_032,SILENT)
      BF(32,0x0000000F,ACT_NOP,rand_031_000,SILENT)
   IE_END(authentication_parameter_rand)

   IE_BEGIN(authentication_parameter_sres)
      BF(32,0x0000000F,ACT_NOP,sres_031_000,SILENT)
   IE_END(authentication_parameter_sres)

   IE_BEGIN(bearer_capability)   /* not filled out yet */
      BF(8,0,ACT_NOP,                         length,SILENT)
      BF(1,0,ACT_NOP,                           ext3,SILENT)
      BF(2,0,ACT_NOP,      radio_channel_requirement,SILENT)
      BF(1,0,ACT_NOP,                coding_standard,SILENT)
      BF(1,0,ACT_NOP,                  transfer_mode,SILENT)
      BF(3,0,ACT_NOP,       info_transfer_capability,SILENT)
      BF(1,0,ACT_NOP,                          ext3a,SILENT)
      BF(1,0,ACT_NOP,                         coding,SILENT)
      BF(2,0,ACT_NOP,                          spare,SILENT)
      BF(4,0,ACT_NOP,      speech_version_indication,SILENT)
      BF(1,0,ACT_NOP,                           ext4,SILENT)
      BF(1,0,ACT_NOP,                         spare2,SILENT)
      BF(2,0,ACT_NOP,                      structure,SILENT)
      BF(1,0,ACT_NOP,                    duplex_mode,SILENT)
      BF(1,0,ACT_NOP,                  configuration,SILENT)
      BF(1,0,ACT_NOP,                           nirr,SILENT)
      BF(1,0,ACT_NOP,                  establishment,SILENT)
      BF(1,0,ACT_NOP,                           ext5,SILENT)
      BF(2,0,ACT_NOP,                access_identity,SILENT)
      BF(2,0,ACT_NOP,                  rate_adaption,SILENT)
      BF(3,0,ACT_NOP,     signalling_access_protocol,SILENT)
      BF(1,0,ACT_NOP,                           ext6,SILENT)
      BF(2,0,ACT_NOP,               layer_1_identity,SILENT)
      BF(4,0,ACT_NOP,     user_info_layer_1_protocol,SILENT)
      BF(1,0,ACT_NOP,                     sync_async,SILENT)
      BF(1,0,ACT_NOP,                          ext6a,SILENT)
      BF(1,0,ACT_NOP,            number_of_stop_bits,SILENT)
      BF(1,0,ACT_NOP,                    negotiation,SILENT)
      BF(1,0,ACT_NOP,            number_of_data_bits,SILENT)
      BF(4,0,ACT_NOP,                      user_rate,SILENT)
      BF(1,0,ACT_NOP,                          ext6b,SILENT)
      BF(2,0,ACT_NOP,              intermediate_rate,SILENT)
      BF(1,0,ACT_NOP,   tx_network_independent_clock,SILENT)
      BF(1,0,ACT_NOP,   rx_network_independent_clock,SILENT)
      BF(3,0,ACT_NOP,                         parity,SILENT)
      BF(1,0,ACT_NOP,                          ext6c,SILENT)
      BF(2,0,ACT_NOP,             connection_element,SILENT)
      BF(5,0,ACT_NOP,                     modem_type,SILENT)
      BF(1,0,ACT_NOP,                           ext7,SILENT)
      BF(2,0,ACT_NOP,               layer_2_identity,SILENT)
      BF(5,0,ACT_NOP,     user_info_layer_2_protocol,SILENT)
   IE_END(bearer_capability)

   IE_BEGIN(channels_needed_for_mobiles_1_and_2)
      BF(2,0,ACT_NOP,second_channel,"spare, any channel")
      BF(2,0,ACT_NOP, first_channel,"spare, any channel")
   IE_END(channels_needed_for_mobiles_1_and_2)

   IE_BEGIN(channel_description)
      BF( 5,M5(0,0,1,0,1),ACT_NOP,          channel_type,"SDCCH/SACCH 4(1)  ")
      BF( 3,            0,ACT_NOP,      time_slot_number,"zero              ")
      BF( 3,          BCC,ACT_NOP,training_sequence_code,"same as BCCH      ")
      BF( 1,            0,ACT_NOP,               hopping,"No                ")
      BF( 2,            0,ACT_NOP,                 spare,SILENT              )
      BF(10,   ARFCN_BCCH,ACT_NOP,                 arfcn,"ARFCN of the BCCH")
   IE_END(channel_description)

   IE_BEGIN(ciphering_key_sequence_number)
      BF(1,        0,ACT_NOP,       spare,SILENT)
      BF(3,M3(1,1,1),ACT_NOP,key_sequence,"no key is available (MS->BS)")
   IE_END(ciphering_key_sequence_number)

   IE_BEGIN(ciphering_key_sequence_number_2)
      BF(1,        0,ACT_NOP,       spare,SILENT)
      BF(3,M3(1,0,1),ACT_NOP,key_sequence,"sent BS->MS")
   IE_END(ciphering_key_sequence_number_2)

   IE_BEGIN(ciphering_mode_setting)
      BF(3,M3(0,0,0),ACT_NOP,algorithm_identifier,"A5/1           ")
      BF(1,        1,ACT_NOP,     start_ciphering,"Start ciphering")
```

```
IE_END(ciphering_mode_setting)

IE_BEGIN(cipher_response)
   BF(3,0,ACT_NOP,          spare,SILENT                         )
   BF(1,0,ACT_NOP,cipher_response,"IMEISV shall not be included")
IE_END(cipher_response)

IE_BEGIN(description_of_the_first_channel_after_time)
   BF(1,0,ACT_STOP,ANONYM,SILENT)             /* not implemented yet, dummy data */
IE_END(description_of_the_first_channel_after_time)

IE_BEGIN(ia_rest_octets)   /* maximum length (11), no hopping, no starting time */
   BF(8,REST_OCTET,ACT_NOP,ANONYM,SILENT) /*  0 */
   BF(8,REST_OCTET,ACT_NOP,ANONYM,SILENT) /*  1 */
   BF(8,REST_OCTET,ACT_NOP,ANONYM,SILENT) /*  2 */
   BF(8,REST_OCTET,ACT_NOP,ANONYM,SILENT) /*  3 */
   BF(8,REST_OCTET,ACT_NOP,ANONYM,SILENT) /*  4 */
   BF(8,REST_OCTET,ACT_NOP,ANONYM,SILENT) /*  5 */
   BF(8,REST_OCTET,ACT_NOP,ANONYM,SILENT) /*  6 */
   BF(8,REST_OCTET,ACT_NOP,ANONYM,SILENT) /*  7 */
   BF(8,REST_OCTET,ACT_NOP,ANONYM,SILENT) /*  8 */
   BF(8,REST_OCTET,ACT_NOP,ANONYM,SILENT) /*  9 */
   BF(8,REST_OCTET,ACT_NOP,ANONYM,SILENT) /* 10 */
IE_END(ia_rest_octets)

IE_BEGIN(iei_63)
   BF(8,0x63,ACT_NOP,ANONYM,SILENT)
IE_END(iei_63)

IE_BEGIN(iei_34)
   BF(8,0x34,ACT_NOP,ANONYM,SILENT)
IE_END(iei_34)

IE_BEGIN(l2_pseudo_length_9)  /* paging request type 1 with TMSI (one mobile)*/
   BF(9,0,ACT_NOP,ANONYM,SILENT)
IE_END(l2_pseudo_length_9)

IE_BEGIN(mobile_allocation)
   BF(8,0,ACT_NOP,length,"length 0 due to hopping disabled")
IE_END(mobile_allocation)

IE_BEGIN(mobile_identity)             /* has 6 octets */
   BF(8,          5,ACT_NOP,  length,"five octets to come")
   BF(4,M4(1,1,1,1),ACT_NOP,  ANONYM,"bits 5 to 8 of octet 3 are '1111'")
   BF(1,          0,ACT_NOP,odd_even,"As applicable for TMSI")
   BF(3,  M3(1,0,0),ACT_NOP,    type,"TMSI")
   BF(4,       0x22,ACT_NOP, digit_2,SILENT)
   BF(4,       0x11,ACT_NOP, digit_1,SILENT)
   BF(4,       0x44,ACT_NOP, digit_4,SILENT)
   BF(4,       0x33,ACT_NOP, digit_3,SILENT)
   BF(4,       0x66,ACT_NOP, digit_6,SILENT)
   BF(4,       0x55,ACT_NOP, digit_5,SILENT)
   BF(4,       0x88,ACT_NOP, digit_8,SILENT)
   BF(4,       0x77,ACT_NOP, digit_7,SILENT)
IE_END(mobile_identity)

IE_BEGIN(mode_of_the_first_channel)
   BF(1,0,ACT_STOP,ANONYM,SILENT)             /* not implemented yet, dummy data */
IE_END(mode_of_the_first_channel)

IE_BEGIN(ms_classmark)
   BF(8,        3,ACT_NOP,                length,SILENT)
   BF(1,        0,ACT_NOP,                    spare,SILENT)
   BF(2,  M2(0,1),ACT_NOP,        revision_level,"phase 2 MS")
   BF(1,        0,ACT_NOP,                es_ind,"No 'Controlled Early Classmark Sending'")
   BF(1,        0,ACT_NOP,                  a5_1,"encryption algorithm A5/1 available    ")
   BF(3,M3(0,0,1),ACT_NOP,   rf_power_capability,"class 2")
   BF(1,        0,ACT_NOP,                spare2,SILENT)
   BF(1,        0,ACT_NOP,         ps_capability,"no pseudo-synchronisation capability")
   BF(2,  M2(0,0),ACT_NOP,ss_screening_indicator,"default value of phase 1")
   BF(1,        0,ACT_NOP,         sm_capability,"no point to point SMS")
   BF(1,        0,ACT_NOP,                   vbs,"no VBS capability or no notification wanted")
   BF(1,        0,ACT_NOP,                  vgcs,"no VGCS capability or no notification wanted")
   BF(1,        0,ACT_NOP,  frequency_capability,"no extention band G1")
   BF(1,        0,ACT_NOP,           classmark_3,"no additional MS capability information")
   BF(5,        0,ACT_NOP,                spare3,SILENT)
   BF(1,        0,ACT_NOP,                  a5_3,"A5/3 not available")
   BF(1,        0,ACT_NOP,                  a5_2,"A5/2 not available")
IE_END(ms_classmark)

IE_BEGIN(p1_rest_octets)   /* pag. req. type1 : 22 - 9 (L2 pseud. len) =  13 bytes */
   BF(8,REST_OCTET,ACT_NOP,ANONYM,SILENT) /*  0 */
   BF(8,REST_OCTET,ACT_NOP,ANONYM,SILENT) /*  1 */
   BF(8,REST_OCTET,ACT_NOP,ANONYM,SILENT) /*  2 */
   BF(8,REST_OCTET,ACT_NOP,ANONYM,SILENT) /*  3 */
   BF(8,REST_OCTET,ACT_NOP,ANONYM,SILENT) /*  4 */
   BF(8,REST_OCTET,ACT_NOP,ANONYM,SILENT) /*  5 */
   BF(8,REST_OCTET,ACT_NOP,ANONYM,SILENT) /*  6 */
   BF(8,REST_OCTET,ACT_NOP,ANONYM,SILENT) /*  7 */
   BF(8,REST_OCTET,ACT_NOP,ANONYM,SILENT) /*  8 */
   BF(8,REST_OCTET,ACT_NOP,ANONYM,SILENT) /*  9 */
   BF(8,REST_OCTET,ACT_NOP,ANONYM,SILENT) /* 10 */
   BF(8,REST_OCTET,ACT_NOP,ANONYM,SILENT) /* 11 */
   BF(8,REST_OCTET,ACT_NOP,ANONYM,SILENT) /* 12 */
IE_END(p1_rest_octets)

IE_BEGIN(page_mode)
   BF(2,0,ACT_NOP,spare,"two spare bits  ")
   BF(2,0,ACT_NOP,pm   ,"Normal Paging")
IE_END(page_mode)
```

```
   IE_BEGIN(power_command)
      BF(1,0,ACT_STOP,ANONYM,SILENT)             /* not implemented yet, dummy data */
   IE_END(power_command)

   IE_BEGIN(rach)
      BF(3,    M3(1,0,0),ACT_CHECK ,establishment_cause,"paging ind. was 'any channel'")
      BF(5,M5(1,1,1,1,1),ACT_NOP,   random_reference,"ignore Random Reference      ")
   IE_END(rach)

   IE_BEGIN(request_reference)
      BF(3,    M3(1,0,0),ACT_NOP,random_access_info,"As in CHAN REQ")
      BF(5,M5(1,1,1,1,1),ACT_NOP,  random_reference,SILENT)
      BF(5,            0,ACT_NOP,               t1_,SILENT)
      BF(6,            0,ACT_NOP,               t3 ,SILENT)
      BF(5,            0,ACT_NOP,               t2 ,SILENT)
   IE_END(request_reference)

   IE_BEGIN(rr_cause)
      BF(1,0,ACT_STOP,ANONYM,SILENT)             /* not implemented yet, dummy data */
   IE_END(rr_cause)

   IE_BEGIN(signal_call_waiting)
      BF(8,M8(0,0,0,0,0,1,1,1),ACT_NOP,signal_value,"(Any non-reserved value)")
   IE_END(signal_call_waiting)

   IE_BEGIN(timing_advance)
      BF(2,0,ACT_NOP,         spare,SILENT)
      BF(6,0,ACT_NOP,timing_advance,"0"   )
   IE_END(timing_advance)

   IE_BEGIN(transaction_identifier_source)
      BF(4,M4(0,0,0,0),ACT_NOP,ANONYM,SILENT)
   IE_END(transaction_identifier_source)

   IE_BEGIN(transaction_identifier_dest)
      BF(4,M4(1,0,0,0),ACT_NOP,ANONYM,SILENT)
   IE_END(transaction_identifier_dest)

/*-------------------------------------------------------------------*\
| Messages
\*-------------------------------------------------------------------*/

MSG3_BEGIN(paging_request_type_1)
   IE(l2_pseudo_length_9)
   IE(skip_indicator)
   IE(rr_management_protocol_discriminator)
   IE(paging_request_type_1_message_type)
   IE(page_mode)
   IE(channels_needed_for_mobiles_1_and_2)
   IE(mobile_identity)
   IE(p1_rest_octets)
MSG3_END(paging_request_type_1)

MSG3_BEGIN(channel_request)
   IE(rach)
MSG3_END(channel_request)

MSG3_BEGIN(immediate_assignment)
   IE(l2_pseudo_length_21)
   IE(skip_indicator)
   IE(rr_management_protocol_discriminator)
   IE(immediate_assignment_message_type)
   IE(page_mode)
   IE(spare_half_octet)
   IE(channel_description)
   IE(request_reference)
   IE(timing_advance)
   IE(mobile_allocation)
   IE(ia_rest_octets)
MSG3_END(immediate_assignment)

MSG3_BEGIN(paging_response)
   IE(skip_indicator)
   IE(rr_management_protocol_discriminator)
   IE(paging_response_message_type)
   IE(ciphering_key_sequence_number)
   IE(spare_half_octet)
   IE(ms_classmark)
   IE(mobile_identity)
MSG3_END(paging_response)

MSG3_BEGIN(authentication_request)
   IE(skip_indicator)
   IE(mobility_management_protocol_discriminator)
   IE(authentication_request_message_type)
   IE(ciphering_key_sequence_number_2)
   IE(spare_half_octet)
   IE(authentication_parameter_rand)
MSG3_END(authentication_request)

MSG3_BEGIN(authentication_response)
   IE(mobility_management_protocol_discriminator)
   IE(skip_indicator)
   IE(authentication_response_message_type)
   IE(authentication_parameter_sres)
MSG3_END(authentication_response)

MSG3_BEGIN(ciphering_mode_command)
   IE(skip_indicator)
   IE(rr_management_protocol_discriminator)
   IE(ciphering_mode_command_message_type)
   IE(ciphering_mode_setting)
   IE(cipher_response)
MSG3_END(ciphering_mode_command)

MSG3_BEGIN(ciphering_mode_complete)
   IE(skip_indicator)
   IE(rr_management_protocol_discriminator)
   IE(ciphering_mode_complete_message_type)
MSG3_END(ciphering_mode_complete)

MSG3_BEGIN(setup)                           /* contains "signal" but no "Bearer Cap" */
   IE(transaction_identifier_source)
   IE(call_control_protocol_discriminator)
   IE(setup_message_type)
   IE(iei_34)
   IE(signal_call_waiting)
MSG3_END(setup)

MSG3_BEGIN(call_confirmed)                  /* contains bearer capability */
   IE(transaction_identifier_dest)
   IE(call_control_protocol_discriminator)
   IE(call_confirmed_message_type)
   IE(bearer_capability)
MSG3_END(call_confirmed)

MSG3_BEGIN(connect)
   IE(transaction_identifier_dest)
   IE(call_control_protocol_discriminator)
   IE(connect_message_type)
MSG3_END(connect)

MSG3_BEGIN(alerting)
   IE(transaction_identifier_dest)
   IE(call_control_protocol_discriminator)
   IE(alerting_message_type)
MSG3_END(alerting)

MSG3_BEGIN(assignment_command)
   IE(skip_indicator)
   IE(rr_management_protocol_discriminator)
   IE(assignment_command_message_type)
   IE(description_of_the_first_channel_after_time)
   IE(power_command)
   IE(iei_63)
   IE(mode_of_the_first_channel)
MSG3_END(assignment_command)

MSG3_BEGIN(assignment_complete)
   IE(skip_indicator)
   IE(rr_management_protocol_discriminator)
   IE(assignment_complete_message_type)
   IE(rr_cause)
MSG3_END(assignment_complete)

MSG3_BEGIN(connect_acknowledge)
   IE(transaction_identifier_source)
   IE(call_control_protocol_discriminator)
   IE(connect_acknowledge_message_type)
MSG3_END(connect_acknowledge)
```

4.4 MTC, dynamic part (TCSL)

```
/*------------------------------------------------------------------*\
| Sample TCSL test case,
| Rename this file to Testcase.h and move it with the other files
| file in this directory to gsm\ms\tds\.
| Then create the testdll.dll with "nmake testdll.mak" in
| gsm\int\msdev\testdll. The outfile is in gsm\int\msdev\testdll\testdll_
\*------------------------------------------------------------------*/

#define  ARFCN_BCCH              122

#define  NCC                     0x5
#define  BCC                     0x6
#define  BSIC                    ((NCC<<3)|(BCC))
#define  RFN                     0

#include "tcsl.h"
#include "tcsl_gsm.h"

#include "10_1_2.h"  /* define system information IEs       */
#include "10_1_4.h"  /* define other information elements   */

  ISS_INIT(1);                              /* allocate 1 BS */

  BS_SET_SYS_INFO ( 0 , system_information_type_1 )
  BS_SET_SYS_INFO ( 0 , system_information_type_2 )
  BS_SET_SYS_INFO ( 0 , system_information_type_3 )
  BS_SET_SYS_INFO ( 0 , system_information_type_4 )
  BS_SET_SYS_INFO ( 0 , system_information_type_5 )
  BS_SET_SYS_INFO ( 0 , system_information_type_6 )

  BS_ON_OFF       ( 0 , TRUE )
  BS_SET_SCH      ( 0 , BSIC , RFN )
  BS_SET_ARFCN    ( 0 , ARFCN_BCCH )
  BS_SET_POWER    ( 0 , -10 )

  BS_MSG3_SEND (0,paging_request_type_1  , SILENT )
  BS_RACH_AWAIT(0,channel_request        , SILENT )
  BS_MSG3_SEND (0,immediate_assignment   , SILENT )
  BS_MSG3_AWAIT(0,paging_response        , SILENT )
  BS_MSG3_SEND (0,authentication_request , SILENT )
  BS_MSG3_AWAIT(0,authentication_response, SILENT )
  BS_MSG3_SEND (0,ciphering_mode_command , SILENT )
  BS_MSG3_AWAIT(0,ciphering_mode_complete, SILENT )
  BS_MSG3_SEND (0,setup                  , SILENT )

  ISS_DEINIT();                             /* clean up */
```

4.5 Configuration File

```
TRX: 0
Time Slot: 0
Channel Combi: 0
Power of MS: 0
training sequence: 0
output power: -40
Timing Advance: 0
Sys info in tranceiver: 0
Hopping?: false
ARFCN: 0
TRX: 1
Time Slot: 1
Channel Combi: 1
Power of MS: 0
training sequence: 0
output power: -40
Timing Advance: 0
Sys info in tranceiver: 0
Hopping?: TRUE
MAIO: 1
HSN: 2
number of frequency: 3
Hop freq1: 1
Hop freq2: 2
Hop freq3: 3
TRX: 2
Time Slot: 2
Channel Combi: 2
Power of MS: 0
training sequence: 0
output power: -40
Timing Advance: 0
Sys info in tranceiver: 0
Hopping?: false
ARFCN: 0
TRX: 3
Time Slot: 3
Channel Combi: 3
```

```
training sequence: 0
output power: -40
Hopping?: false
ARFCN: 0
Ncc: 0
Bcc: 0
BS-PA-MFRMS: 0
BS-AG-BLKS-RFS: 0
```

5 Discussion

Some extensions of TCSL are discussed in this section.

5.1 Recursive bit fields

The static part of TCSL has a three level hierarchy expressed by (1) bit fields, (2) information elements and (3) Layer 3 messages. This hierarchy was chosen to reflect the data structures in the GSM 04.08 specification. One way to generalize this approach is to remove information elements and Layer 3 messages from TCSL and to integrate them into bit fields.
Still a method is required to define bit fields that contain more than 32 bits, .e.g.

```
BF_BEGIN(big_bf)
       BF ( 10,          0, ACT_NOP, bf_1, SILENT )
       BF ( 20,    0x12345, ACT_NOP, bf_2, SILENT )
       BF ( 30, 0x3FFFFFFF, ACT_NOP, bf_3, SILENT )
BF_END(big_bf)

BF_BEGIN(bigger_bf)
       BF ( 16,          0, ACT_NOP, bf_4, SILENT )
       BF (  0,     big_bf, ACT_NOP, bf_5, SILENT )
BF_END(bigger_bf)
```

Instead of a constant initial value for **bf_5**, a reference to another bit field is used. This "reference type" is indicated by a length of "**0**". The BF macro must be changed and probably a C type cast must be applied to force the reference into the value component of the bit field.
This recursive definition of bit fields allows any level of data hierarchies. information elements and Layer 3 messages can be removed from TCSL.

5.2 Ranges

Bit field ranges are not available in TCSL. Ranges should be used in AWAIT operation to allow a more detailed comparison of bit fields. As currently implemented TCSL allows only a simple equal/unequal comparison. One approach to allow "greater than" or "less than" would be:

```
RANGE_BEGIN ( range_1 )
       RANGE( lo_bound_1 , up_bound_1 )
       RANGE( lo_bound_2 , up_bound_2 )
       RANGE( lo_bound_3 , up_bound_3 )
RANGE_END ( range_1 )

BF_BEGIN ( bf_10 )
       BF_RANGED ( 23, range_1, ACT_NOP, bf_1, SILENT )
BF_END ( bf_10 )

AWAIT ( bf_10 )
```

The value of bf_10 after a successful execution of AWAIT should meet the following constraints:

lo_bound_1 <= value_of(bf_10) <= up_bound_1 or

```
lo_bound_2 <= value_of(bf_10) <=  up_bound_2 or
lo_bound_3 <= value_of(bf_10) <=  up_bound_3 or
```

Appendices

A. Acronyms

DS-WCDMA	Direct Sequence/Spread Wideband Code Division Multiple Access

B. Glossary

International Mobile Telecommunication 2000 (IMT-2000/ITU-2000)	Formerly referred to as FPLMTS (Future Public Land-Mobile Telephone System), this is the ITU's specification/family of standards for 3G. This initiative provides a global infrastructure through both satellite and terrestrial systems, for fixed and mobile phone users. The family of standards is a framework comprising a mix/blend of systems providing global roaming. <URL: http://www.imt-2000.org/>