Technical Documentation

LOCOSTO DSP/MCU INTERFACE FOR ROM CODE 3801

Document Number:	L1_GS814_1
TI Department	European Wireless Terminal Chipset Business Unit
Version:	1.0
Status:	Approved
Date:	17-Aug-05

IMPORTANT NOTICE

Texas Instruments Incorporated and / or its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products, software and services at any time and to discontinue any product, software or service without notice. Customers should obtain the latest relevant information during product design and before placing orders and should verify that such information is current and complete.

All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment. TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI products, software and / or services. To minimize the risks associated with customer products and applications, customers should provide adequate design, testing and operating safeguards.

Any access to and / or use of TI software described in this document is subject to Customers entering into formal license agreements and payment of associated license fees. TI software may solely be used and / or copied subject to and strictly in accordance with all the terms of such license agreements.

Customer acknowledges and agrees that TI products and / or software may be based on or implement industry recognized standards and that certain third parties may claim intellectual property rights therein. The supply of products and / or the licensing of software do not convey a license from TI to any third party intellectual property rights and TI expressly disclaims liability for infringement of third party intellectual property rights.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products, software or services are used.

Information published by TI regarding third–party products, software or services does not constitute a license from TI to use such products, software or services or a warranty, endorsement thereof or statement regarding their availability. Use of such information, products, software or services may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Change History

Date of change	Changed by	Approved by	Date of approval	Version	Notes
Feb-03-2005	Vinayak Bellur			0.1	(1)
Feb-03-2005	Saurabh Gupta	Sebastien Guiriec		0.2	(2)
Jun-03-2005	Sebastien Guiriec			0.3	(3)
Jun-24	K. Bala			0.4	(4)
Jul-13-2005	Sebastien Guiriec	Sebastien Guiriec		0.5	(5)
Aug-3-2005	K. Bala	Sebastien Guiriec	Aug-3-2005	1.0	(6)

Notes

(1) Creation
(2) Update for DRP/DCO/Triton
(3) Update according to new DCO
(4) Update for APC changes
(5) Update for C-Port API
(6) Addition of d_thr_usf_detect and putting the document is approved status.

Table of Contents

List of Figures and Tables

List of References

Reference and name	Title
[L1D_GS813]	L1D_GS813 ROM 38 description
[L1_AS041]	MP3 - MCU-DSP interface
[L1_AS131_2]	High frequency I2S protocol audio output – MCU-DSP Interface of the BGD task used for tests
[L1_MS351_1]	SWH MCU-DSP API Definition
[L1_MS361_1]	DCO – MCU/DSP Interface
[L1_AS111_1]	ANR MCU/DSP interface
[L1D_AS021]	TTY – API interface
[L1_AS261_1]	ES - MCU/DSP Interface

1 Introduction

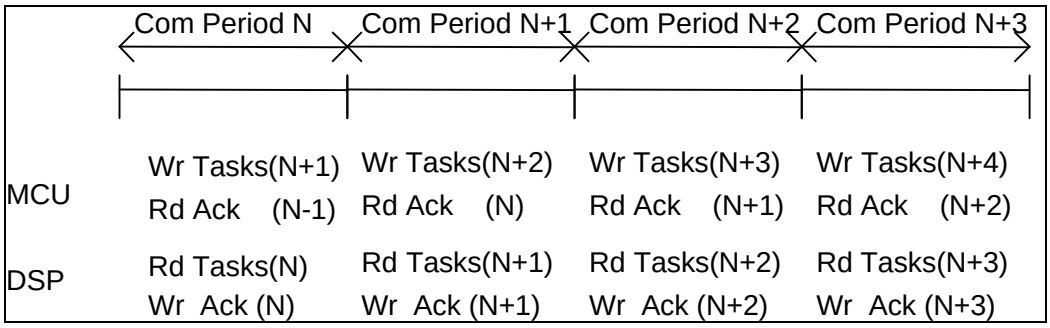
This specification describes the MCU/DSP interface for Locosto devices and its future derivatives.

For more details about different ROMs refers to **Error! AutoText entry not defined.**.

2 INTRODUCTION

2.1 Communication period

A communication period is typically one TDMA period or a multiple value of a TDMA period. While MCU is writing the commands (tasks) with associated data for the next communication period, DSP is executing tasks and writing corresponding results (acknowledge) with associated data for the current communication period. The communication period is stamped thanks to an interrupt available for MCU and DSP (Figure Here below illustrates pipeline operation of communication).



2.2 Memory mapping

MCU (ARM) and DSP (LEAD) communicate via a Dual Port memory connected to the API (ARM parallel interface). The memory area used to communicate is mapped in the data space and is accessed on a word basis (16bits).

Within this "common memory", three areas have been defined according to the periodicity of the data updating:

* **Non Double Buffered Area (NDB)**:	contain data that may not be accessed on a TDMA period. These data will be valid for (at least) several TDMAs (e.g. TCH information buffers).
* **Double Buffered Area (DB)**:	consists in Communication Buffers that contain data changed on a TDMA period by MCU or DSP (e.g. tasks identifiers, power measurements). Communication Buffers are accessed by both MCU and DSP in an asynchronous way during a communication period. It is the reason why they are "*double buffered*".

The MCU should only access to the memory areas reserved in API for MCU/DSP interface. The other areas in API are used by DSP to perform acquisitions with RF or to perform signal processing (static and working variables)

2.3 Double buffering

To support the communication period mechanism, the Communication Buffer is double buffered. It is divided in two parts:

MCU_to_DSP part: This part contains the two buffers that are used for MCU to DSP transfers (tasks with datas): BufMD_0 and BufMD_1.

DSP_to_MCU part: This part contains the two buffers that are used for DSP to MCU transfers (acknowledges with datas): BufDM_0 and BufDM_1.

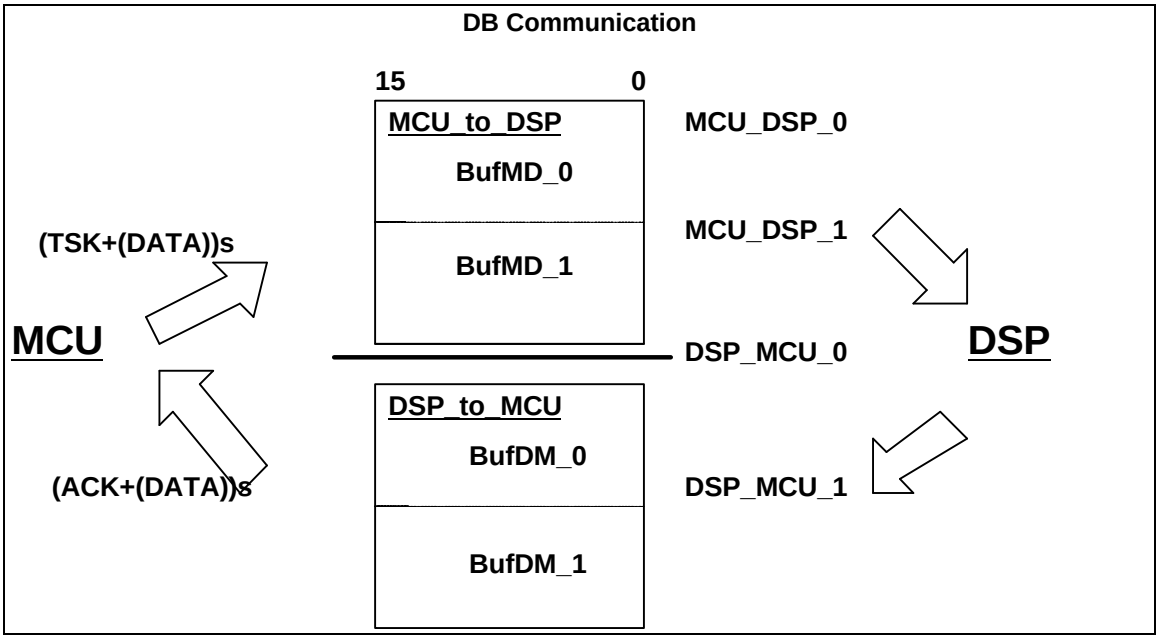


Table here below shows how buffers are used during consecutive periods:

	Com_period(n)	Com_perod(n+1)	Com_period(n+2)	.....
MCU	W0/R0	W1/R1	W0/R0	.....
DSP	R1/W1	R0/W0	R1/W1	.....

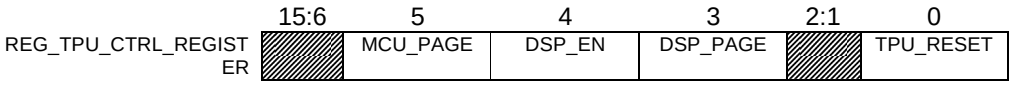
W0: Write in CB buffer 0 (BufDM_0 for DSP, BufMD_0 for MCU)
W1: Write in CB buffer 1 (BufDM_1 for DSP, BufMD_1 for MCU)

R0: Read in CB buffer 0 (BufMD_0 for DSP, BufDM_0 for MCU)
R1: Read in CB buffer 1 (BufMD_1 for DSP, BufDM_1 for MCU)

Switching of Communication buffers on each consecutive period is not mandatory, since the choice of the buffer is determined by MCU.

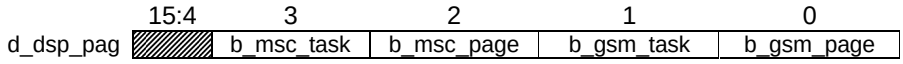
2.4 DSP-MCU DB Page synchronization

The MCU knows its current working page by reading the TPU register REG_TPU_CTRL_REGISTER:



Bit	Active level	Observation
TPU_RESET	1	Written by MCU to reset TPU: DSP_PAGE = 0, MCU_PAGE = 0, DSP_EN = 0
DSP_PAGE	0 for page 0 1 for page 1	TPU copies MCU_PAGE in DSP_PAGE at Communication Interrupt.
DSP_EN	1 for new scenario	Set by MCU to enable a Communication Interrupt. Cleared by TPU on occurrence of Communication Interrupt.
MCU_PAGE	0 for page 0 1 for page 1	Read by MCU to select available page. MCU_PAGE = /MCU_PAGE on occurrence of Communication Interrupt.

This register is not used for protocol, the page control register *d_dsp_pag* from the API (PARAM area) is used instead. This register is written by MCU when a new Communication Interrupt is activating the DSP. It is read by the DSP at Communication Interrupt and stored into the local variable *d_dsp_page_copied* to allow the MCU to modify it. Then the MCU/DSP communication buffer space in the API is fixed by software with the variable *d_dsp_page_copied*.



Bit	Active level	Observation
b_gsm_page	0 for page 0 1 for page 1	Indicate the page for GSM Tasks.
b_gsm_task	1	Communication Interrupt corresponds to a GSM Task.
b_msc_page	0 for page 0 1 for page 1	Not currently used.
b_msc_task	1	Communication Interrupt corresponds to a MISC Task.

The DSP uses the bit *b_gsm_task* to know if GSM tasks have been requested to the current TDMA. The bit *b_gsm_page* indicate the page number for GSM tasks. The DSP know if misc tasks (Tone/keybeep, voice memo, melody, speech recognition) are requested for the current TDMA by reading the *b_msc_task* bit. Any combination of *b_gsm_task* and *b_msc_task* can occur. The *b_msc_page* bit is not currently used.

3 DB INTERFACE DEFINITION

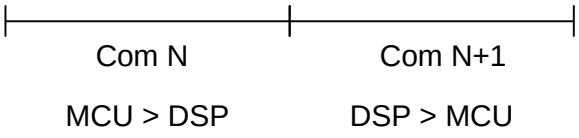
3.1 Header of Double Buffering Messages (DB)

The message structures for the various operating modes are here below described. For each task, a brief functional description is given as well as communication period and number of periods in master process. The DSP will process the tasks (if any) in the following order:
1 - Downlink task
2 - Uplink task
3 - Measurement or Monitoring task

Some fixed addresses have been dedicated to Reception, Transmission and Measurement/Monitoring tasks written in communication buffer.

When some tasks are missing (e.g. uplink task in Cell Selection), the task is loaded at **NO_TASK** command. The numbers of the downlink and uplink bursts are read according to the task ordered (e.g. there is no need of a number for the downlink bursts of a neighbor cell). The measurement task identifier consists in the number of measurements to perform within a communication period (**0,1, 2** or **3** measurements).

Representation shows MCU to DSP commands and corresponding DSP to MCU acknowledge. If MCU to DSP commands are executed in communication period N, DSP to MCU acknowledge will be available in communication period N+1:



3.2 Mapping of Double Buffers (DB) – Common to GSM and GPRS

GSM means that the data is accessed during the TDMAs used in GSM circuit switched mode (paging, dedicated modes..)
GPRS means that the data is accessed during the TDMAs used in GPRS mode (idle and transfer).
MSC means that the data is accessed during non-GSM TDMAs (keybeeps in (packet) idle/(packet) dedicated mode, voice memo operations…)

Note concerning MS:
- Alone, these annotations mean that both GSM and GPRS modes use the associated API area.
- With (GSM) or (GPRS) subscript annotation, these mean that only the indicated mode access to the associated API area.

3.2.1 MCU -> DSP doubled buffer interface

Offset	Name		Description
0	**d_task_d**	GSM	Index of the downlink task of the current TDMA {17..37} NBN_TASK, EBN_TASK, NBS_TASK, EBS_TASK, NP_TASK, EP_TASK,CB_TASK, DDL_TASK, ADL_TASK, TCHT_TASK, TCHA_TASK Bit 15 indicates if the I/Q data must be swapped before being demodulated.
1	**d_burst_d**	GSM	Index of the downlink burst for CCH {0..3} Bit 15 indicates if the I/Q data must be swapped before being demodulated.
2	**d_task_u**	GSM	INDEX OF THE UPLINK TASK OF THE CURRENT TDMA {11..14} AUL_TASK, DUL_TASK, TCHT_TASK,TCHA_TASK Bit 15 indicates if the I/Q data must be swapped before being transmitted to the analog baseband uplink converter.
3	**d_burst_u**	GSM	Index of the uplink burst for CCH {0..3} Bit 15 indicates if the I/Q data must be swapped before being transmitted to the analog baseband uplink converter.
4	**d_task_md**	GSM GPRS	Index of the monitoring task of the current TDMA {1..9} ð GSM: 0 : No task 1: PM1_TASK 2 : PM2_TASK 3 : PM3_TASK 4 : PM4_TASK 5 : FB_TASK 6 : SB_TASK 7 : SB_NEW_TASK 8 : TCH_FB_TASK 9 : TCH_SB_TASK 10-32 : No task 33 : TST_VER ð GPRS: 0-4 : No task 5 : FB_TASK 6 : SB_TASK 7 : SB_NEW_TASK 8 : TCH_FB_TASK 9 : TCH_SB_TASK

			10-16 : No task 17 : BCCH Neighbour monitoring 18-99 : No task 100 : No task 101 : 1 Interference measurement … 108 : 8 Interference measurements bit 15 : I/Q swap enable bit for FB, SB and BCCH activities.
5	**d_backgroun d**	GSM	Not used.
6	**d_debug**	GSM GPRS	relative FN to the reset time of the MS ('time stamp')
7	**d_task_ra**	GSM	code for RACH {10} RACH_TASK Bit 15 indicates if the I/Q data must be swapped before being transmitted to the analog baseband uplink converter.
8	**d_fn_report**	GSM	two concatenated bytes giving the FN%26 8 msbs gives FN%104 ([0-103] range) and are used by DSP to compute the SID position in TCH mode. This byte is called b_fn_sid. 8 lsbs gives a "frame number" modulo 104 normalized to reporting period. It is used by DSP to assert SACCH position in the TCH mode. This byte is called b_fn_report
9	**d_ctrl_tch**	GSM	bit field control for the current TCH Bit0_Bit3: b_chan_mode bits are specifying the TCH mode: 00 signaling only. 01 tch/full rate speech 02 tch/half rate speech 03 tch/full rate data 9600 04 tch/full rate data 4800 06 tch/full rate data 2400 08 tch/enhanced full rate 09 tch/full rate data 14400 10 tch AMR half rate speech 11 tch AMR full rate speech Bit4-Bit5 :b_chan_type bits specify the type of traffic channel: 00 invalid (channel is not allocated) 01 tch/full rate and tch/enhanced full rate 02 tch half rate Bit6: b_reset_sacch: this bit indicates to the DSP to reset the SACCH Bit7: b_vocoder_on: vocoder is enabled (ensure b_sync_tch = 1 to reset the vocoder and the deinterleaver). Bit8: b_sync_tch: Uplink/Downlink Start Request of all TCH/F and TCH/H channel mode Bit9: b_sync_amr: AMR change request. To set to 1 when new AMR parameters (ACS, ICMI, THRS, …) are available in the NDB API. Bit10:b_stop_tch_ul: Uplink Stop Request Bit11:b_stop_tch_dl: Downlink Stop Request Bit12-Bit14: b_tch_loop bits specify loop mode: 000: No Loop 001: Loop A 010: Loop B 011: Loop C 100: Loop D (TCH/HR) 101: Loop E (TCH/HR) 110: Loop F (TCH/HR)

			111 : Loop I (TCH AFS/AHS) Bit15: b_subchannel : indicates the sub-channel number of the current half-rate speech channel .
10	**reserved**	GSM	reserved for DSP modules debugging and extensions
11	**d_ctrl_abb**	GSM GPRS	ð GSM: Bit field indicating the DRP/ Wrapper registers to send (circuit switched GSM) bit 0: b_ramp: the ramp information (a_ramp[] is located in NDB). bit 1: unused bit 2: b_apcdel2 delays-register in NDB bit 3: b_apcdel delays-register in NDB bit 4 : b_afc : freq control register in DB bit 5..8: unused bit 9..10: reserved (must be set to 0) bit 11..15: unused ð GPRS: bit 0..3: unused bit 4 : b_afc : freq control register in DB bit 5..15 : unused
12	**a_a5fn1** **(old name)** **a_a5_count[0]** **(new name)**	GSM	ciphering parameter 1 (frame number) `| 0 | 0 | 0 | 0 | 0 | T3(5) .... T3(0) | T2(4) ....T2(0)|`
13	**a_a5fn2** **(old name)** **a_a5_count[1]** **(new name)**	GSM	ciphering parameter 2 (frame number) `| 0 0 0 0 0 T1(10) . . . T1(0)|` T1(0) = lsb of T1, T1(10) = msb of T1 T3(0) = lsb of T3, T3(5) = msb of T3 T2(0) = lsb of T2, T2(4) = msb of T2
14	**d_power_ctrl**	GSM	power control in the format of the register in DRP wrapper
15	**d_afc**	GSM GPRS	AFC control value in DRP in Q 14.0 format
16	**d_ctrl_system**	GSM GPRS	*bit field (TSC, BCCH/HOP, abort_bit)* *bit0-bit2: b_tsq :training sequence number* *bit 3 : b_bcch_freq_ind: This bit is set when current TDMA is hopping on the beacon frequency. In this case even if the soft decision found are good the burst could be a dummy burst and d_dem_dummy_flag is set to 1 in DSP demodulation algorithm.* *bits 4..13: unused* *bit 14 : b_strong_abort: Reserved for a strong abort mechanism which allows performing more initialization than normal abort. This bit is only evaluated when b_task_abort is set to 1. Current DSP code does nothing.* *bit 15 : b_task_abort: When this bit is set, the DSP resets the list of pending tasks.*
17	**d_swh_ApplyWhitening_db**	GSM	This variable command dsp whether the whitening inside SWH should be applied or not. Refer **[L1_MS351_1]**
18	**reserved**	GSM	reserved for DSP modules debugging and extensions
19	**reserved**	GSM	reserved for DSP modules debugging and extensions

3.2.2 DSP -> MCU doubled buffer interface

Offset	Name		Description
0	**d_task_d**	GSM	copy of the d_task_d done after receiving TDMA ISR
1	**d_burst_d**	GSM	copy of the d_burst_d done after receiving TDMA ISR
2	**d_task_u**	GSM	copy of the d_task_u done after receiving TDMA ISR
3	**d_burst_u**	GSM	copy of the d_burst_u done after receiving TDMA ISR
4	**d_task_md**	GSM/GPRS	copy of the d_task_md done after receiving TDMA ISR
5	**reserved**	GSM	reserved for DSP modules debugging and extensions
6	**d_debug**	GSM/GPRS	copy of the d_debug done after receiving TDMA ISR
7	**reserved**	GSM	reserved for DSP modules debugging and extensions
8	**d_toa**	GSM/GPRS	d_toa: estimated time of arrival of the burst in bit. The range is [0..9] for NB and [0..49] for SB
9	**d_pm**	GSM/GPRS	d_pm: estimated power measurement of the burst (dBd format) range is ([0-84]*64) then d_pm range is [0 -5376]
10	**d_angle**	GSM/GPRS	d_angle : estimated frequency error of the burst angle value for carrier deviation (-pi,+pi) range as f1.15 format
11	**d_snr**	GSM/GPRS	d_snr : estimated signal to noise ratio of the burst in f6.10 format
12	**d_pm1**	GSM	value of the first monitoring power measurement After execution of TST_VER (computation of DSP checksum) d_pm1 contains the embedded software version number1 of the DSP code.
13	**d_pm2**	GSM	value of the second monitoring power measurement After execution of TST_VER , checksum of DSP code
14	**d_pm3**	GSM	value of the third monitoring power measurement After execution of TST_VER , software version number2
15	**d_pm4/SCH 1**	GSM/GPRS	SCH monitoring demodulation header (GSM idle mode) After execution of the PM4_TASK , SCH1 contains the value of the fourth monitoring power measurement Note: d_pm4 is not used for GPRS.
16	**d_pm5/SCH 2**	GSM/GPRS	SCH monitoring accumulated metric After execution of the PM5_TASK , SCH2 contains the value of the fourth monitoring power measurement Note: d_pm4 is not used for GPRS.
17	**d_pm6/SCH 3**	GSM/GPRS	SCH monitoring estimated number of errors After execution of the PM6_TASK , SCH3 contains the value of the fourth monitoring power measurement Note: d_pm6is not used for GPRS.
18	**d_pm7/SCH 4**	GSM/GPRS	SCH monitoring data frame 1 After execution of the PM7_TASK , SCH4 contains the value of the fourth monitoring power measurement Note: d_pm7s not used for GPRS.
19	**d_pm8/ SCH5**	GSM/GPRS	SCH monitoring data frame 2 After execution of the PM8_TASK , SCH5 contains the value of the fourth monitoring power measurement Note: d_pm8is not used for GPRS.

Note:
Into layer-1 software, SCH1 to SCH5 are also named *a_sch*[5].

3.2.3 New MCU -> DSP doubled buffer interface

Offset	Name		Description
0	**d_dco_algo_ctrl_nb**	GPRS/ GSM	This word tells the DC offset algorithm to be used for normal and synchro burst each RX window. For details refer to **[L1_MS361_1]**
1	**d_dco_algo_ctrl_sb**	GPRS/ GSM	This word tells the DC offset algorithm to be used for power burst each TDMA. For details refer to **[L1_MS361_1]**
2	**d_dco_algo_ctrl_pw**	GPRS/ GSM	This word tells the DC offset algorithm to be used for power burst each TDMA. For details refer to **[L1_MS361_1]**
3	reserved		reserved for DSP modules debugging and extensions
4	reserved		reserved for DSP modules debugging and extensions
5	reserved		reserved for DSP modules debugging and extensions
6	reserved		reserved for DSP modules debugging and extensions
7	reserved		reserved for DSP modules debugging and extensions
8	reserved		reserved for DSP modules debugging and extensions
9	reserved		reserved for DSP modules debugging and extensions
10	reserved		reserved for DSP modules debugging and extensions
11	reserved		reserved for DSP modules debugging and extensions
12	reserved		reserved for DSP modules debugging and extensions
13	reserved		reserved for DSP modules debugging and extensions
14	reserved		reserved for DSP modules debugging and extensions
15	reserved		reserved for DSP modules debugging and extensions

3.2.4 Synchronization of the AFC programmation

When the AFC is to be programmed (indicated by b_afc in DB) the DSP will send the formatted value to the DRP register whose address is indicated in the d_drp_afc_add_api variable. For DRP the configuration of the AFC will be done at the beginning of the TDMA.

3.3 Mapping of Double Buffers (DB) – GPRS only

The Trace DB is used to latch the trace pointer on each TDMA interrupt to be able to dump dynamically some DSP trace information.

3.3.1 Trace MCU -> DSP doubled buffer interface

Offset	Name		Description
0	reserved		reserved for DSP modules debugging and extensions

3.3.2 Trace DSP -> MCU doubled buffer interface

Offset	Name		Description
	d_debug_ptr_begin	GSM / GPRS	Address of the begin of the DSP trace for the previous TDMA
0	d_debug_ptr_end	GSM ? GPRS	Address of the end of the DSP trace for the previous TDMA

3.4 Mapping of Double Buffers (DB) – GPRS only

3.4.1 MCU -> DSP doubled buffer interface

GPRS EXTENSION (TDMA granularity)		
d_task_d_gprs	GPRS	GPRS Downlink activity for the TDMA. This word contains the following bit field: bit 0 : Set to 1 to indicate that RX is enabled for Timeslot 7. bit 1 : Set to 1 to indicate that RX is enabled for Timeslot 6. … bit 7 : Set to 1 to indicate that RX is enabled for Timeslot 0. bit 8-13 : Unused. bit 14 : b_ptcch_d: PTCCH/D is requested if set to 1. bit 15 : I/Q swap enable bit for RX activity.
d_task_u_gprs	GPRS	GPRS Uplink activity for the TDMA. This word contains the following bit field: bit 0 : Set to 1 to indicate that TX is enabled for Timeslot 7. bit 1 : Set to 1 to indicate that TX is enabled for Timeslot 6. … bit 7 : Set to 1 to indicate that TX is enabled for Timeslot 0. bit 8-12 : Unused. bit 13 : b_access_prach: PRACH in access phase if set to 1. bit 14 : b_ptcch_u: PTCCH/U is requested if set to 1. bit 15 : I/Q swap enable bit for TX activity.
d_task_pm_gprs	GPRS	GPRS Power monitoring activity for the TDMA. This word contains the following bit field: bit 0 : Set to 1 to indicate that PM is enabled for Timeslot 7. bit 1 : Set to 1 to indicate that PM is enabled for Timeslot 6. … bit 7 : Set to 1 to indicate that PM is enabled for Timeslot 0. bit 8-15 : Unused.
d_burst_nb_gprs	GPRS	Index of the burst into the bloc: Range [0 to 3]. This register can be used for uplink or downlink.

GPRS EXTENSION (Timeslot granularity)		
a_ctrl_abb_gprs [8]	GPRS	bit field indicating the DRP / wrapper registers to update in GPRS mode: bit 0: b_ramp: the ramp a_ramp2_gprs_x[]) is located in NDB. bit 1: unused bit 2: b_apcdel2 delays-register in NDB. bit 3: b_apcdel delays-register in NDB bit 4 : unused bits 5:7: number of the ramp to be used (up to 8 ramps). This index is not linked to the timeslot information. bit 8 : b_ms_rule: this bit is used by Layer-1 to request a specific re-programation for the corresponding timeslot, especially to load a ramps with multislot Tx configuration. bit 9: b_apcdel_bis, used to program APCDEL1 register two times per TDMA. bit 10: b_apcdel2_bis, used to program APCDEL2 register two times per TDMA. Note: power (d_ctrl_power_gprs) is automatically updated by the scheduler on each timeslots, so it does not need to be controlled by the MCU.
a_ctrl_power_gprs [8]	GPRS	power control value

Note: First array element corresponds to timeslot 0 and last array element corresponds to timeslot 7.

3.4.2 DSP -> MCU doubled buffer interface

GPRS EXTENSION (TDMA granularity)		
d_task_d_gprs	GPRS	Copy of d_task_d_gprs done after receiving TDMA ISR.
d_task_u_gprs	GPRS	Copy of d_task_u_gprs done after receiving TDMA ISR.
d_task_pm_gprs	GPRS	Copy of d_task_pm_gprs done after receiving TDMA ISR.
d_burst_nb_gprs	GPRS	Copy of d_burst nb_gprs done after receiving TDMA ISR.

GPRS EXTENSION (Timeslot granularity)		
a_burst_toa_gprs [8]	GPRS	Demod - time of arrival
a_burst_pm_gprs [8]	GPRS	Demod – power measurement or neighbor cell power measurement
a_burst_angle_gprs [8]	GPRS	Demod – angle
a_burst_snr_gprs [8]	GPRS	Demod – snr

Notes:

- First array element corresponds to timeslot 0 and last array element corresponds to timeslot 7.
- When the timeslot is dedicated to a power measurement, the result is written into *a_burst_pm_gprs.*

4 NDB INTERFACE DEFINITION

GSM means that the data is accessed during the TDMAs used in GSM mode (paging, dedicated modes..)
GPRS means that the data is accessed during the TDMAs used in GPRS mode (idle and transfer)
MSC means that the data is accessed during non-GSM TDMAs (keybeeps in idle/dedicated mode, voice memo operations…)

Note concerning MSC:

- Alone, these annotations mean that both GSM and GPRS modes use the associated API area.
- With (GSM) or (GPRS) subscript annotation, these mean that only the indicated mode access to the associated API area.

4.1 Mapping of the NDB area – Common the GSM and GPRS

Name		Description
a_cd[15]	GSM	CCCH/SACCH demodulation block
a_fd[15]	GSM	FACCH demodulation block
a_dd_0[22]	GSM/MSC	TCH subchannel 0 demodulation block
a_dd_1[22]	GSM/MSC	TCH subchannel 1 demodulation block
a_cu[15]	GSM	CCCH/SACCH uplink block
a_fu[15]	GSM	FACCH uplink block
a_du_0[22]	GSM/MSC	TCH subchannel 0 uplink block
a_du_1[22]	GSM/MSC	TCH subchannel 1 uplink block
d_rach	GSM	RACH information (format below)
d_a5_mode	GSM/GPRS	Ciphering algorithm selection: 0: no ciphering, 1: A5/1, 2: A5/2, 3: A5/3 **WARNING: the value 0 is mandatory for GPRS.**
d_tch_mode	GSM/GPRS	d_tch_mode:bit field control register Bit0: Reserved. Must be set to 0. Bit1 : unused Bit2: b_dtx DTX mode selection (0:no DTX) Bit3: b_play_ul: when set to 1 , the DSP is taking uplink information to channel encoder from a_du_x (x=NOT(b_sub_channel)) Bit4: Reserved. Must be set to 0. Bit5: b_voice_memo_dtx: Specifies DTX mode for voice memo feature **ONLY** (Full-rate and AMR). 0: No DTX 1: DTX Bit6: b_vocoder_select: Selects the operations of the Voice Memo dictaphone 0: use the TCH/FS vocoder like in previous implementations of the Voice Memo 1: use the AMR vocoders and the corresponding DSP internal scheduling Bit7-Bit10: d_guards: reserved for rampup control (default value=0), numbers of guard bits is ((d_guards + 4)and 15) Bit11: b_pcm_select : copy PCM samples during voice memo FR to PCM API buffer. Bit12: Not used Bit13: Disable DMA FIFO emptying Loop Bit14: Mute audio path downlink

Name		Description
		Bit15: Mute audio path uplink
a_drp_ramp[20]	GSM	power ramp up/down
d_version_number1	GSM/GPRS	corresponds to DSP ROM CODE version number
d_version_number2	GSM/GPRS	corresponds to DSP PATCH version number
d_apcdel	GSM/GPRS	corresponds to APCDEL
d_apclev	GSM/GPRS	Corresponds to APCLEV register of the APC module. bit 0-14: Value provided by MCU bit 15: New value to program to the APCLEV register. Reset by DSP before programming
d_apcctrl2	GSM/GPRS	Corresponds to APCCTRL2 register of the APC module. bit 0-14: Value provided by MCU bit 15: New value to program to the APCCTRL2 register. Reset by DSP before programming
d_dai_on_off	GSM	Bit0 : read-acknowledge bit set by the MCU and reseted by the DSP DAI mode is in the bits 11 and 12: 00 : No DAI operation (MCSI interface shut down) 01 : Test of speech decoder/DTX functions (downlink) 10 : Test of speech encoder/DTX functions (uplink) 11 : Test of acoustic devices and A/D and D/A (voice path)
d_error_status	GSM/GPRS/MSC	Bit field for DSP status returned to MCU ð GSM bit 0: Error during a RHEA bus access bit 1: unused bit 2: the DMA is still waiting I/Q samples in the beginning of a TDMA bit 3: error pushing the DMA programmation task (DMA interrupts did not occurred) bit 4: error pushing the downlink DMA task in the Task Queue (DMA interrupts did not occurred) bit 5: error pushing the downlink DMA task in the Pending Task Queue (DMA interrupts did not occurred) bit 6: unused bit 7: error in VoiceMemo operation: underflow of data from the MCU during playing the memo. bits 8-15: unused bits 11: Stack overflow indication bit. bits 12-15: unused ð GPRS bit 0: unused bit 1: unused *bit 2: the DMA is still waiting I/Q samples in the beginning of a TDMA* *bit 3: error pushing the DMA programmation task (DMA interrupts did not occurred)* *bit 4: error pushing the downlink DMA task in the Downlink Task Queue (DMA interrupts did not occurred)* *bit 5: error pushing the downlink DMA task in the Pending Task Queue (DMA interrupts did not occurred)* *bit 6:* unused *bit 7:* reserved

Name		Description
		bit 8: error pushing the uplink task in the Uplink Task Queue (TPU_PROG interrupts did not occurred) *bit 9: error pushing the uplink task in the Pending Task Queue* *bit 10: error pushing task(not from uplink or downlink queue) in the Pending Task Queue* *bit 11: Stack overflow indication bit.* bits 12-15: unused
d_dsp_pag	GSM/GPRS	Used for DSP-MCU DB page synchronization. Refer to chapter 2.4.
d_audio_init	GSM/GPRS/MSC	Bit0 : unused Bit1 : used to activate the internal audio loop with the audio filters activated (samples from the microphone are looped back to the loudspeaker) Bits 2-3: reserved Bit 4: TTY initialization Bit 5-15 : reserved
d_audio_status	GSM/GPRS/MSC	Read only bits for information about the audio loop activity. Bit0 : unused Bit1: used to indicate the activity of the internal audio loop Bits 2-15 : reserved
d_ra_conf	GSM	Data services configuration bit field WARNING: This word must never be overlaid and must be set to 0 if IDS is not used.
d_ra_act	GSM	Data services configuration bit field
d_ra_test	GSM	Data services configuration bit field
d_ra_statu	GSM	Data services status bit field
d_ra_stad	GSM	Data services status bit field
d_fax	GSM	Data services configuration bit field
a_rau/a_ntu/a_fau	GSM	Data services buffer uplink (Fax transmissions)
a_rad/a_ntd/a_fad	GSM	Data services buffer downlink (Fax transmissions)
d_pll_config	GSM/GPRS	Latency after IDLE3 wake-up. Bits 15: force IDLE 2 (never go to IDLE 3) Bits 14: force IDLE 1(never go to IDLE 2/3) Bit 0-11:Latency before leaving IDLE 3 mode in DSP cycles (minimum value 1, max 0xFFF). .
d_software_version1	GSM/MSC	Loaded by the DSP at reset time. Contains in the MSB byte the version number of the ROM code and in the LSB byte the revision number of the ROM code.
d_software_version2	GSM/MSC	Loaded by the DSP at reset time with 0 without patch and with a MSB indicating the revision of the patch, the LSB byte must be 0.
d_apcdel2	GSM/GPRS	Corresponds to APCDEL2
d_drp_afc_add_api	GSM/GPRS	contains the address where AFC value need to be written.
d_mcsi_select	GSM/GPRS/MMS-AMR	Selection of MCSI port for DAI or BT configuration (must be set to 0) 0 : MCSI-1 port is selected
d_audio_ctrl	GSM/GPRS	Control the audio algorithms Bit 0-3 : audio_loop_ctrl Bit 4 : iir_start Bit 5 : limiter_start

Name		Description
		Bit 6 : anr_start
d_audio_state	GSM/GPRS	Status register for the audio algorithms Bit 0-3 : audio_loop_state Bit 4 : iir_state Bit 5 : limiter_state Bit 6 : anr_state

4.1.1 Background task

Name		Description
d_background_enable	GSM/GPRS	Background tasks enable bit field. The MCU sets the corresponding bit to 1 to enable the task.
d_background_abort	GSM/GPRS	Background task abort (stop and initialize) bit field. The MCU sets the corresponding bit to 1 to abort the task. The DSP reset the bit as an acknowledge.
d_background_state	GSM/GPRS	The DSP returns the state of tasks. Task is active if corresponding bit is set to 1.
d_max_background	GSM/GPRS	The MCU indicates the maximum number of background task into the range [0 to 16].
a_background_tasks [16]	GSM/GPRS	The MCU configures the background task manager with up to 16 background tasks.
a_back_task_io [16]	GSM/GPRS	The MCU indicates the address of the I/O structure of up to 16 different background tasks.
d_dsp_state	GSM/GPRS	Report the state of the DSP to the MCU: 0: DSP is currently running 1: DSP is in Idle 1 mode 2: DSP is in Idle 2 mode 3: DSP is in Idle 3 mode

4.1.2 Dynamic download

The following variables are related to dynamic download MCU/DSP interface.

Name		Description
d_api_dwl_download_ctrl	GSM/GPRS	Control word.
d_api_dwl_error_code	GSM/GPRS	Error code.
d_api_dwl_function_address	GSM/GPRS	Address of the download area or of the install-uninstall function.
d_api_dwl_crc	GSM/GPRS	CRC of the downloaded code.
d_api_dwl_size	GSM/GPRS	Size of the download area.
d_api_dwl_write_pointer	GSM/GPRS	Running write pointer address in DSP address format

4.1.3 DSP Trace

Name		Description
p_debug_buffer	GSM/GPRS	DSP Debug trace: Address of the DSP write pointer
d_debug_buffer_size	GSM/GPRS	DSP Debug trace: Size of the Data field
d_debug_trace_type	GSM/GPRS	DSP Debug trace: Requested features Bit 3-0: Trace level bit 0: ISR level bit 1: Basic level bit 2: Kernel level bit 3: Unused Bit 7-4: Optional features

Name		Description
		bit 4: Buffer header bit 5: Buffer bit 6: Burst bit 7: Timer Bit 11-8: Unused Bit 14-12: Request type 0: New level and/or feature configuration 1: New buffer address and/or size… current log could be lost 2: Same as 1 but with a dump of current data log into the new buffer Bit 15: New request flag (set by the ARM, reset by the DSP)

4.1.4 Modem modules

All the next chapters describe the parameters link to MODEM activities.

4.1.4.1 Frequency burst/Synchronization burst

Name		Description
d_fb_det	GSM/GPRS	flag value is 1 if frequency burst search attempt is successful
d_fb_mode	GSM/GPRS	Frequency burst detection bandwidth 0 at startup (acquisition), 1 otherwise (tracking)
d_fb_toa / a_sync_demod[0]	GSM/GPRS	FB/SB scenarios return TOA. The L1 software uses the name a_sync_demod[0].
d_fb_pm / a_sync_demod[1]	GSM/GPRS	FB/SB scenarios return PM. The L1 software uses the name a_sync_demod[1].
d_fb_angle / a_sync_demod[2]	GSM/GPRS	FB/SB scenarios return ANGLE. The L1 software uses the name a_sync_demod[2].
d_fb_snr / a_sync_demod[3]	GSM/GPRS	FB/SB scenarios return SNR. The L1 software uses the name a_sync_demod[3].
a_sch[5]/ a_sch26 [4]	GSM/GPRS	Buffer for SB decoded block. SCH demodulation block (format below) used during TCH monitoring

4.1.4.2 DCO

The DCO module has been modified for DRP integration. For more information about the description of the MCU/DSP API refers to **[L1_MS361_1]**.

Name		Description
d_dco_algo_mode	GSM/GPRS	This word tells the mode of operation to be observed by the DSP. Bit0: Manual (0) or script assisted (1) Initialized once during init time.
d_dco_samples_per_symbol	GSM/GPRS	Number of samples per symbol (IQ pair). It could be 1, 2, or 4. Used in manual and script assisted mode. Initialized once during init time.
a_dco_calc_addr_i[4]	GSM/GPRS	Address of the accumulated DC value for I channel for 4 bursts. The address has to be a DSP address. For manual mode MCU loads the ROC_CALC_I hardware register's address. Values in these addresses are 32-bit Initialized once during init time.
a_dco_calc_addr_q[4]	GSM/GPRS	Address of the accumulated DC value for Q channel for 4 bursts. The address has to be a DSP address. For manual mode MCU loads the ROC_CALC_Q hardware register's address. Values in these addresses are 32-bit. Initialized once during init time.
d_dco_num_sample_nb	GSM/GPRS	Number of samples for which DC has been accumulated for NB
d_dco_num_sample_sb	GSM/GPRS	Number of samples for which DC has been accumulated for SB
d_dco_num_sample_pw	GSM/GPRS	Number of samples for which DC has been accumulated for PW burst
d_dco_csf_attenuation	GSM/GPRS	It is the attenuation of the DC signal which is accumulated at resampler output when it passes through ROC, IQMC, ZIF and CSF blocks. It is a number which lies between 0 and 1 and is scaled by 2^16. (Unsigned fixed format 0.16)
d_dco_fcw	GSM/GPRS	Frequency control word. Initialized once during init time. It is the same as the FCW field (bits 3 to 10) of the ZIF_CW register of DRP. Fs = (1.083 MHz) Fs*(FCW/512) FCW = 0x01 => 2.115234375 KHz FCW = 0x2F => 99.69140625 KHz FCW = 0xFF => 539.384765625 KHz
d_dco_starting_offset_int	GSM/GPRS	Integer part of the starting phase offset for burst 0 in terms of index in the lookup table. It lies between 0-512. (The sine wave table is of size 512). Initialized once during init time.
d_dco_starting_offset_fraction	GSM/GPRS	Fractional part of the starting phase offset for burst 0 in terms of index in the lookup table. It lies between 0 and 1 and is scaled by 2^16. Initialized once during init time. (Unsigned fixed format 0.16)

4.1.5 Acoustic modules

This section describes all the acoustic modules applicable during all GSM/GPRS activities when some audio applications are running on the 8 KHz audio path.

4.1.5.1 FIR uplink

Name		Description
FIR31_uplink	GSM/GPRS/MSC	FIR coefficients for the audio uplink path in Q2.14 format. For example the floating point value 0.5 is coded 0x2000. The FIR filtering is followed by an amplification controled by the register "d_audio_gain_ul"

4.1.5.2 IIR downlink

Name		Description
d_iir_input_scaling	GSM/GPRS/MSC	Used to scale the input at entry of the IIR to avoid overflows inside the IIR.
d_iir_fir_scaling	GSM/GPRS/MSC	Used to scale the output of the IIR before using the FIR to avoid any scaling in the FIR.
d_iir_input_gain_scaling	GSM/GPRS/MSC	The scaling factor applied at input of the filter for the global gain.
d_iir_output_gain_scaling	GSM/GPRS/MSC	Scaling factor at the output of the filter for the gain.
d_iir_output_gain	GSM/GPRS/MSC	A gain between [0,1] to tune the value of the global gain applied by the module.
d_iir_feedback	GSM/GPRS/MSC	Used to tune the rounding noise of the IIR implementation and to remove the bias.
d_iir_nb_iir_blocks	GSM/GPRS/MSC	Describes the number of blocks for the given implementation of the systolic lattice IIR filter.
d_iir_nb_fir_coefs	GSM/GPRS/MSC	Number of coefficients for the FIR (degree+1 of the FIR polynomial).
a_iir_iir_coefs	GSM/GPRS/MSC	Contains the coefficients of the IIR lattice filter.
a_iir_fir_coefs	GSM/GPRS/MSC	Contains the coefficients of the FIR filter.
a_lim_mul_low	GSM/GPRS/MSC	Second and third multiplying coefficients for the low frequency signal.

4.1.5.3 Limiter downlink

Name		Description
a_lim_mul_high	GSM/GPRS/MSC	Second and third multiplying coefficients for the high frequency signal.
d_lim_gain_fall_q15	GSM/GPRS/MSC	Gain falling
d_lim_gain_rise_q15	GSM/GPRS/MSC	Gain rising
d_lim_block_size	GSM/GPRS/MSC	Number of samples in an input block.
d_lim_nb_fir_coefs	GSM/GPRS/MSC	Number of coefficients in the filter. It must be an odd number.
d_lim_slope_update_period	GSM/GPRS/MSC	Number of samples between each update of the limiter slope.
a_lim_filter_coefs	GSM/GPRS/MSC	Coefficients for the FIR filter.

4.1.6 Audio Application

4.1.6.1 MP3

The following variables are related to MP3 decoder MCU/DSP API. For more information about MP3 API refers to **[L1_AS041]**.

Name		Description
d_mp3_api_header	GSM/GPRS	Header.
d_mp3_api_channel	GSM/GPRS	DMA channel number.
d_mp3_api_init	GSM/GPRS	Init command.
d_mp3_api_play	GSM/GPRS	Play command.
d_mp3_api_pause	GSM/GPRS	Pause command.
d_mp3_api_restart	GSM/GPRS	Restart command.
d_mp3_api_stop	GSM/GPRS	Stop command.
d_mp3_api_end	GSM/GPRS	End command.
d_mp3_api_request_index	GSM/GPRS	Index for requested data.
d_mp3_api_request_size	GSM/GPRS	Size of requested data.
d_mp3_api_provided_size	GSM/GPRS	Size of data provided.
d_mp3_api_error_code	GSM/GPRS	Error code.

4.1.6.2 CPORT test

The following variables are related to C-PORT test background task. For more information about this feature refers to. **[L1_AS131_2]**.

Name		Description
d_cport_api_dma_install	GSM/GPRS	Control word to install a DMA callback. Refer to
d_cport_api_dma_channel	GSM/GPRS	DMA channel number.
d_cport_api_dma_rootcause	GSM/GPRS	Rootcause updated by the DMA callback.

4.1.6.3 Tones/VoiceMemo/Melody/keybeeps

Name		Description
d_toneskb_init	MSC (Tones/Keybeep)	Tones & voice memo control (see below)
d_toneskb_status	MSC (Tones/Keybeep)	Tones & voice memo status bits (see below)
d_k_x1_tX[3]	MSC (Tones)	
d_pe_rep	MSC (Tones)	
d_pe_off	MSC (Tones)	
d_se_off	MSC (Tones)	
d_bu_off	MSC (Tones)	
d_t0_on	MSC (Tones)	
d_t0_off	MSC (Tones)	
d_t1_on	MSC (Tones)	
d_t1_off	MSC (Tones)	
d_t2_on	MSC (Tones)	
d_t2_off	MSC (Tones)	
d_k_x1_kt0	MSC (Keybeep)	
d_k_x1_kt1	MSC (Keybeep)	
d_dur_kb	MSC (Keybeep)	
d_shift_dl	GSM/GPRS/MSC	Gain control for the DL audio path in the voice memo record, in Q8.8 format (0dB = 0x0100) This gain control is effective in Dedicated mode Voice memo operations only.
d_shift_ul	GSM/GPRS/MSC	Gain control for the UL audio path in the voice memo record, in Q8.8 format (0dB = 0x0100) This gain control is effective in Idle, GPRS and Dedicated mode Voice memo operations.
d_melo_ctrl	MSC (E0 & E1)	Reserved for E0 & E1 Melody generator. See S811 documentation.
d_melo_max_used	MSC (E0 & E1)	Reserved for E0 & E1 Melody generator. See S811 documentation.
melo[8][4]	MSC (E1)	Reserved for Melody generator. See S811 documentation. • E1 format uses up to 8 oscillators.
d_melo_select	MSC (E0 & E1)	0x0002: Melody E1
d_audio_gain_ul	MSC	Register used to add (d_audio_gain_ul+1)*6dB gain on the audio uplink path
d_amms_ul_voc	GSM/GPRS/ MMS-AMR	Bits0-2: b_amms_channel_type: Selects the AMR codec to be used during the dictaphone recording.

Note: Tones control words are not defined as GSM or GPRS words, because tones are considered as a generic tasks (not telecommunication task).

4.2 Mapping of the NDB area – GSM only

Name		Description
a_kc[4]	GSM	64 bit ciphering key a_kc[0].0 = lsb, a_kc[3].15 = msb

Name		Description
d_amrschd_api_amr_1	GSM/ AMR	MCU/DSP AMR API – compressed word 1 Bit 0 : d_icmi_dl Bit 1-2 : d_icm_dl Bit 3 : d_icmi_ul Bit 4-5 : d_icm_ul Bit 6 : d_nscb Bit 7-15 : spare
d_amrschd_api_amr_2	GSM/ AMR	MCU/DSP AMR API – compressed word 2 Bit 0-7 : d_acs_ul Bit 8-15 : d_acs_dl
d_amrschd_api_amr_3	GSM/ AMR	MCU/DSP AMR API – compressed word 3 Bit0-5: d_thr1_dl Bit 6-11 : d_thr2_dl Bit 12-15: d_hyst3_dl
d_amrschd_api_amr_4	GSM/ AMR	MCU/DSP AMR API – compressed word 4 Bit0-3: d_hyst1_dl Bit 4-7 : d_hyst2_dl Bit 8-13 : d_thr3_dl Bit 14 : d_nsync Bit 15 : spare
a_amrschd_raatsch_ul[6]	GSM/ AMR	MCU/DSP buffer for AMR RAATSCH UL
a_amrschd_raatsch_dl[6]	GSM/ AMR	MCU/DSP buffer for AMR RAATSCH DL
d_amrschd_snr_est	GSM/ AMR	Output value of the estimated SNR for debug purpose
d_amrschd_thr_onset_afs	GSM/ AMR	Threshold for ONSET detection in AFS
d_amrschd_thr_sid_first_afs	GSM/ AMR	Threshold for SID FIRST detection in AFS
d_amrschd_thr_ratscch_afs	GSM/ AMR	Threshold for RATSCCH detection in AFS
d_amrschd_thr_update_afs	GSM/ AMR	Threshold for SID UPDATE detection in AFS
d_amrschd_thr_onset_ahs	GSM/ AMR	Threshold for ONSET detection in AHS
d_amrschd_thr_sid_ahs	GSM/ AMR	Threshold for SID frames detection in AHS
d_amrschd_thr_ratscch_marker	GSM/ AMR	Threshold for RATSCCH MARKER detection in AHS
d_amrschd_thr_snr_no_data	GSM/ AMR	Threshold for SNR used for SPEECH degraded rejection
a_d_macc_thr_afs[8]	GSM	d_macc decision thresholds for AFS. Speech degraded is rejected when the d_macc value is lower than it. [0] : AFS 4.75 [1] : AFS 5.15 [2] : AFS 5.90 [3] : AFS 6.70 [4] : AFS 7.40 [5] : AFS 7.95 [6] : AFS 10.2 [7] : AFS 12.2
a_d_macc_thr_ahs[6]	GSM	d_macc decision thresholds for AHS. Speech degraded is rejected when the d_macc value is lower than it. [0] : AHS 4.75

Name		Description
		[1] : AHS 5.15 [2] : AHS 5.90 [3] : AHS 6.70 [4] : AHS 7.40 [5] : AHS 7.95
a_drp_ramp[20]	GSM	Power ramp up/down in the DRP registers format.
a_a5_kc[8]	GSM	**For algorithm A5/1 or A5/2, cipher key length is fixed at 54 bits.** **For A5/3 algorithm, a 128-bit input key is programmed, which is a combination of cipher Kc key.** The 128-bit pseudo-key Ck is formed from Kc and written into this buffer. For 64 bit cipher key Kc, Ck = Kc || Kc For 128 bit cipher key Kc, CK[127] … CK[0] = Kc[127] … Kc[0] For key length > 64 bits and < 128 bits, CK[127] … CK[128-KLEN] = Kc[KLEN-1] … Kc[0] CK[127-KLEN] … CK[0] = Kc[KLEN-1] … Kc[2*KLEN-128] where KLEN = cipher key length (Kc) in bits.
d_swh_SaicFlag	GSM	**For details refer to document L1_MS351_1.doc** Values : 0 à SWH disabled 1 à SWH Enabled

4.2.1 Acoustic modules

The next sections describe the MCU/DSP parameters of the acoustic modules available during GSM speech communication only (TCH/FS, TCH/EFS, TCH/HS,TCH/AFS and TCH/AHS only)

4.2.1.1 AEC uplink

This module is used to remove the acoustic echo on the uplink path.

Name		Description
d_ctrl_aec	GSM	Echo canceller & Speech enhancement control bit 0 : acknowledge bit set by the MCU and reseted by the DSP after reading d_aec_ctrl bit 1 : enables the AEC module bit 2 : unused bit 3-4 : unused bit 5-6 : unused bit 7 : reset trigger of AEC set by the MCU and reseted by the DSP bit 8 : unused bit 9 : unable the copy of AEC-VAD output parameters into the API for debug bits 10-15 : unused **Warning:** when entering into GPRS mode, the DSP scheduler disables AEC module.
d_es_level_api	GSM	Additional suppression factor added on the uplink audio path when downlink audio energy is detected. Default value is -12 dB = 0x1FFF. 0dB is coded with 0x7FFF.
d_mu_api	GSM	Filter adaptation step for the echo canceller, default value is 0x5000

4.2.1.2 ANR uplink

The ANR module is used to reduce the background noise on the uplink path. For more information of the MCU/DSP API refers to **Error! AutoText entry not defined..**

Name		Description
d_anr_min_gain	GSM	AGC reference gains (noise: -12 dB, speech : 0 dB)
d_anr_vad_thr	GSM	Threshold for connection of VAD and spectral subtraction
d_anr_gamma_slow	GSM	Gain slow attenuation factor
d_anr_gamma_fast	GSM	Gain fast attenuation factor
d_anr_gamma_gain_slow	GSM	No more used
d_anr_gamma_gain_fast	GSM	No more used
d_anr_thr2	GSM	VAD decision thresholds
d_anr_thr4	GSM	VAD decision thresholds
d_anr_thr5	GSM	VAD decision thresholds
d_anr_mean_ratio_thr1	GSM	VAD decision mean ratio
d_anr_mean_ratio_thr2	GSM	VAD decision mean ratio
d_anr_mean_ratio_thr3	GSM	VAD decision mean ratio
d_anr_mean_ratio_thr4	GSM	VAD decision mean ratio
d_anr_div_factor_shift	GSM	Used to control variation of gain

4.2.1.3 Echo Suppressor

This module is used to control the residual echo where AEC is unable to cancel the echo in the non-idle acoustic environment. It works both in UL and DL.

Name		Description
d_es_mode	GSM	Bit map for enabling/disabling the ES. For more details refer to **[L1_AS261_1]**
es_gain_dl	GSM	Receive loss compensation. This value is in Q11 format.
es_gain_ul_1	GSM	Coupling loss compensation. This value is in Q11 format.
es_gain_ul_2	GSM	Near-end propagation loss compensation. This value is in Q11 format.
d_es_tcl_fe_ls_thr	GSM	TCL reference threshold in far-end mode for loud signals. This value is in Q15 format.
d_es_tcl_dt_ls_thr	GSM	TCL reference threshold in double-talk mode for loud signals. This value is in Q15 format
d_es_tcl_fe_ns_thr	GSM	TCL reference threshold in far-end mode for nominal signals. This value is in Q15 format
d_es_tcl_dt_ns_thr	GSM	TCL reference threshold in double-talk mode for nominal signals. This value is in Q15 format
d_es_tcl_ne_thr	GSM	TCL reference threshold in near-end mode. This value is in Q15 format
d_es_ref_ls_pwr	GSM	TCL reference threshold in near-end mode. This value is in Q15 format
d_es_switching_time	GSM	Switching time value in milliseconds
d_es_switching_time_dt	GSM	Double-talk switching time value in milliseconds
d_es_hang_time	GSM	Hangover time for switching
a_es_gain_lin_dl_vect[0]	GSM	Downlink attenuation level in idle state. Format is Q15.
a_es_gain_lin_dl_vect[1]	GSM	Downlink attenuation level in double-talk state. Format is Q15.
a_es_gain_lin_dl_vect[2]	GSM	Downlink attenuation level in far-end state. Format is Q15.
a_es_gain_lin_dl_vect[3]	GSM	Downlink attenuation level in near-end state. Format is Q15.
a_es_gain_lin_ul_vect[0]	GSM	Uplink attenuation level in idle state. Format is Q15.
a_es_gain_lin_ul_vect[1]	GSM	Uplink attenuation level in double-talk state. Format is Q15.
a_es_gain_lin_ul_vect[2]	GSM	Uplink attenuation level in far-end state. Format is Q15.
a_es_gain_lin_ul_vect[3]	GSM	Uplink attenuation level in near-end state. Format is Q15.

4.2.2 Audio Application

The next sections describe the MCU/DSPparameters of the audio application available during GSM speech communication only (TCH/FS, TCH/EFS, TCH/HS,TCH/AFS and TCH/AHS only)

4.2.2.1 TTY/GTT

This module is used to permit deaf, hard of hearing, and speech-impaired persons to use TTY services on mobile phone. For more information refers to **Error! AutoText entry not defined.**.

Name		Description
a_TTY_FIFO_1	GSM	TTY Fifo: from DSP TTY estimator to MCU Baudot decoder. Refer to AS021 for further details.
a_TTY_FIFO_2	GSM	TTY Fifo: from MCU CTM transmitter to DSP CTM modulator.
a_TTY_FIFO_3	GSM	TTY Fifo: from DSP CTM estimator to MCU CTM receiver.
a_TTY_FIFO_4	GSM	TTY Fifo: from MCU Baudot encoder to DSP TTY modulator.
d_tty_status	GSM	Status register for the process.
d_tty_detec_thres	GSM	Control register for the TTY threshold.
d_ctm_detec_shift	GSM	Control register for the CTM threshold
d_tty_fa_thres	GSM	Control register for the TTY false detection
d_tty_mod_norm	GSM	Control the output tty modulator amplitude.
d_tty_reset_buffer_ul	GSM	Control the enable of the buffer ul process reset for double characters.
d_tty_loop_ctrl	GSM	Control the enable for the debug mode loop process in the CTM or TTY side.
p_tty_loop_buffer	GSM	Initialize the intermediate address buffer in case of the loop mode is enabled.

4.3 Mapping of the NDB area – GPRS only

GPRS EXTENSION (Timeslot granularity)		
a_ctrl_ched_gprs [8]	GPRS	bit field control for the current timeslot: bit0_bit3: b_chan_mode bits are specifying the PTCH mode: 0 Not applicable 1 Downlink, CS auto-detect. 2 CS1 downlink with CS1 forced 3 Not applicable 4 CS2 downlink with CS2 forced 5 CS3 downlink with CS3 forced 6 CS4 downlink with CS4 forced 7 to 08 Not applicable
a_ul_buffer_gprs [8]	GPRS	Uplink data buffer number for the uplink scenario. Range: [0 to 7] for data buffers, [8 to 15] for polling buffers.
a_usf_gprs [8]	GPRS	Expected USF into the range [0 to 7]
d_apcdel_bis	GPRS	Same as d_apcdel but only used when b_apcdel_bis is set into a_ctrl_abb_gprs[TS]. Used to program the APCDEL1 register with up to two different values during the same TDMA.
d_apcdel2_bis	GPRS	Same as d_apcdel2 but only used when b_apcdel2_bis is set into a_ctrl_abb_gprs[TS]. Used to program the APCDEL2 register with up to two different values during the same TDMA.

Note: First array element corresponds to timeslot 0 and last array element corresponds to timeslot 7.

GPRS EXTENSION (TDMA granularity)		
d_sched_mode_gprs	GSM/GPRS	bit field for scheduler control: Bit 0: b_switch_to_gprs, switch from circuit switched to GPRS mode Bit 1: b_switch_to_gsm, switch from GPRS to circuit switched mode. These two bits are active only if b_task_abort bit (from d_ctrl_system) is set. Bits 0-1 are reset by the DSP as acknowledge of the switch. Bits 2-3: b_mac_mode 0 Dynamic allocation (default) 1 Extended dynamic allocation 2 Fixed allocation, not half bupplex mode 3 Fixed allocation, half duplex mode

GPRS EXTENSION (TDMA granularity)		
		Bit 4: unused Bit 5: b_rif_rx_mode is used for Downlink buffer management. Two possible algorithms can be used: algorithm 156.25 is used if 0 (default). If set to 1, the algorithm 157 is used. Note: Two BTS may use different algorithms, then this bit is located into NDB area and it is sampled when b_task_abort bit is set in d_ctrl_system by L1. Bit 6: b_loop_mode is used to enable loop from the output of the channel decoder to the input of the channel encoder. This loop mode is the one defined by R&S for the GPRS test mode without protocol layers. This mode is not compliant with ETSI 11.10 specification.
d_usf_updated_gprs	GPRS	bit field that indicates the USF decoding results to the MAC-S layer Two bits are used for each timeslot. These two bits have the following meaning: 00 – invalid 01 – valid and good USF 10 – valid and bad USF bits 0:1 are used for timeslot 7, bits 2:3 for timeslot 6, …
d_win_start_gprs	GPRS	The BTS physical timeslot number of the first Rx or Tx slot. This number is changed only after a L1-frame re-synchronization.This word is sampled when the b_task_abort bit is set in d_ctrl_system by L1.
d_usf_vote_enable	GPRS	Bitfield used to support USF granularity: Bit7 correspond to Rx on TS0. … Bit0 correspond to Rx on TS7. When a bit is set to 1, the DSP has the responsibility to manage the USF mechanism for the corresponding timeslot. When a bit is set to 0, the DSP is completely controlled by the Layer-1 for the corresponding timeslot.
a_interference_meas_gprs[8]	GPRS	Interference measurement results.
a_ptcchu_gprs[4]	GPRS	PTCCH uplink block
a_dd_md_gprs[16]	GPRS	PTCCH downlink block
a_drp_ramp2_gprs_0[20] **a_drp_ramp2_gprs_1[20]** **a_drp_ramp2_gprs_2[20]** **a_drp_ramp2_gprs_3[20]** **a_drp_ramp2_gprs_4[20]** **a_drp_ramp2_gprs_5[20]** **a_drp_ramp2_gprs_6[20]** **a_drp_ramp2_gprs_7[20]**	GPRS	Power ramp up/down in the DRP registers format. There are up to 8 ramps available.
d_rlcmac_rx_no_gprs	GPRS	This word is located just before the first

GPRS EXTENSION (TDMA granularity)		
		RLC/MAC downlink buffer. This word is not used by DSP.

GPRS EXTENSION (Multislot class dependant)		
a_du_gprs [C_NB_TX_MAX][29]	GPRS	PDTCH or PRACH or PACCH uplink block. C_NB_TX_MAX is the maximum number of transmit slots. It depends on the mobile class (see below)
a_pu_gprs [C_NB_TX_MAX][15]	GPRS	Poll response buffers (CS-1 or PRACH) C_NB_TX_MAX is the maximum number of transmit slots. It depends on the mobile class (see below)
a_dd_gprs [C_NB_RX_MAX][31]	GPRS	All non PTCCH/D downlink block C_NB_TX_MAX is the maximum number of receive slots. It depends on the mobile class (see below)

In order to reduce the API memory, the number of RLC/MAC buffers is optimized in accordance with the targeted multislot class mobile. The rest of API is designed to support all multislot class mobile defined by ETSI. This approach permits to reduce the mapping modification between two multislot class mobiles.

The following table presents the number of RLC/MAC buffers required for two typical multislot class mobiles:

	MS class 1	MS class 12
a_du_gprs	1	4
a_pu_gprs	1	4
a_dd_gprs	1	4

Refer to the chapter 10.1.1.2 for further information on buffer management.

5 PARAM INTERFACE DEFINITION

Only GPRS parameters and HW dedicated parameters have been added in this chapter.

Name		Description
d_transfer_rate	GSM/GPRS	
d_lat_mcu_bridge	GSM/GPRS	Latency after has set CLKOUT_HIGH to 1.
d_lat_mcu_hom2sam	GSM/GPRS	Latency after has set API_HOM to 0 and before DSP enters in SAM.
d_lat_mcu_bef_fast_access	GSM/GPRS	MCU must wait before switching to faster access in HOM mode.
d_lat_dsp_after_sam	GSM/GPRS	DSP must not access API right after change.
p_gprs_install_address	GSM/GPRS	Patch install address for GSM and GPRS: • Without any patch, set 0x7002. • To install patch, set 0x015E.
d_dmacc_threshold	GPRS	Parameter for CS4 channel decoder.
d_sd_threshold	GPRS	Parameter for CS4 channel decoder.
d_nb_max_iterations	GPRS	Parameter for CS4 channel decoder.
d_gsm_bgd_mgt	GSM	Bit-field to choose between the classic GSM scheduling and the new background task scheduling: Bit0: unused Bit1: unused Bit2-15: Classic GSM scheduling is used when bit is set to 0. New background task scheduling is used when bit is set to 1.
d_cn_sw_workaround	GSM/GPRS	Configuration of the software work-around: Bit0 : Work-around to support CRTG. Bit1 : DMA reset on critical DMA still running cases, refer to REQ01260.

6 DETAILS ON THE INTERFACE – MCU/DSP Address and Offset

6.1 Address list in API – DB areas

6.1.1 doubled buffer interface (DB)

Variables	MCU Address	DSP Address	Offset in bytes
d_mcu_dsp0	0xFFD0 0000	0x0800	0000
d_mcu_dsp1	0xFFD0 0028	0x0814	0028
d_dsp_mcu0	0xFFD0 0050	0x0828	0050
d_dsp_mcu1	0xFFD0 0078	0x083C	0078

6.1.2 New MCU -> DSP doubled buffer interface

Variables	MCU Address	DSP Address	Offset in bytes
d_db_mcu_dsp_new0	0xFFD0 0760	0x0BB0	0760
d_db_mcu_dsp_new1	0xFFD0 0780	0x0BC0	0770

Detail address of the variable inside the new MCU -> DSP DB:

Name	MCU address		DSP address	
	DB0	DB1	DB0	DB1
d_dco_algo_ctrl_nb	0xFFD0 0760	0xFFD0 0780	0x0BB0	0x0BC0
d_dco_algo_ctrl_sb	0xFFD0 0762	0xFFD0 0782	0x0BB1	0x0BC1
d_dco_algo_ctrl_pw	0xFFD0 0764	0xFFD0 0784	0x0BB2	0x0BC2

6.1.3 doubled buffer interface (DB) – (GPRS Only)

Variables	MCU Address	DSP Address	Offset in bytes
d_mcu_dsp0_gprs	0xFFD0 00A0	0x0850	00A0
d_mcu_dsp1_gprs	0xFFD0 00C8	0x0864	00C8
d_dsp_mcu0_gprs	0xFFD0 00F0	0x0878	00F0
d_dsp_mcu1_gprs	0xFFD0 0138	0x089C	0138

6.1.4 doubled buffer interface (DB) – (Trace Only)

Variables	MCU Address	DSP Address	Offset in bytes
d_mcu_dsp_trace0	0xFFD0 0180	0x08C0	0180
d_mcu_dsp_trace1	0xFFD0 0182	0x08C1	0182
d_dsp_mcu0_gprs	0xFFD0 0184	0x08C2	0184
d_dsp_mcu1_gprs	0xFFD0 0188	0x08C4	0188

Detail address of the variable inside the new MCU -> DSP DB:

Name	MCU address		DSP address	
	DB0	DB1	DB0	DB1
d_debug_ptr_begin	0xFFD0 0184	0xFFD0 0188	0x08C2	0x08C4
d_debug_ptr_begin	0xFFD0 0186	0xFFD0 018A	0x08C3	0x08C5

6.2 Address list in API – NDB Common to GSM and GPRS

NOTE: The rows shaded in **yellow** color are the changed variables/offset with respect to C+

6.2.1 Control

Variables	MCU Address	DSP Address	Offset in bytes
d_dsp_pag	0xFFD0 01A8	0x08D4	01A8
d_error_status	0xFFD0 01AA	0x08D5	01AA
d_tch_mode	0xFFD0 01AE	0x08D7	01AE
d_version_number1	0xFFD0 01B4	0x08DA	01B4
d_version_number2	0xFFD0 01B6	0x08DB	01B6
d_pll_config	0xFFD0 01BC	0x08DE	01BC
d_apcdel	0xFFD0 01E0	0x08F0	01E0
d_apclev	0xFFD0 01E2	0x08F1	01E2
d_apcctrl2	0xFFD0 01E4	0x08F2	01E4
d_dai_on_off	0xFFD0 01E8	0x08F4	01E8
a_sch/a_sch26	0xFFD0 01FC	0x08FE	01FC
a_dd_1	0xFFD0 02B0	0x0958	02B0
a_du_1	0xFFD0 02DC	0x096E	02DC
a_cd	0xFFD0 03A4	0x09D2	03A4
a_fd	0xFFD0 03C2	0x09E1	03C2
a_dd_0	0xFFD0 03E0	0x09F0	03E0
a_cu	0xFFD0 040C	0x0A06	040C
a_fu	0xFFD0 042A	0x0A15	042A
a_du_0	0xFFD0 0448	0x0A24	0448
d_rach	0xFFD0 0474	0x0A3A	0474
d_ra_conf	0xFFD0 047E	0x0A3F	047E
d_ra_act	0xFFD0 0480	0x0A40	0480
d_ra_test	0xFFD0 0482	0x0A41	0482
d_ra_statu	0xFFD0 0484	0x0A42	0484
d_ra_statd	0xFFD0 0486	0x0A43	0486
d_fax	0xFFD0 0488	0x0A44	0488
a_rau/a_ntu/a_fau/a_tstu	0xFFD0 048A	0x0A45	048A
a_rad/a_ntd/a_fad/a_tstd	0xFFD0 04B4	0x0A5A	04B4
d_dsp_state	0xFFD0 01C4	0x08E2	01C4
d_mcsi_select	0xFFD001CE	0x8E7	0x1CE

6.2.2 AQI manager

Variables	MCU Address	DSP Address	Offset in bytes
d_cntr_es_ctrl	0xFFD01A82	0x1541	1A82
d_cntr_anr_ul_ctrl	0xFFD01A84	0x1542	1A84
d_cntr_aec_ul_ctrl	0xFFD01A86	0x1543	1A86
d_cntr_agc_ul_ctrl	0xFFD01A88	0x1544	1A88
d_cntr_iir_dl_ctrl	0xFFD01A92	0x1549	1A92
d_cntr_lim_dl_ctrl	0xFFD01A94	0x154A	1A94
d_cntr_agc_dl_ctrl	0xFFD01A98	0x154C	1A98
d_aqi_status	0xFFD01A9E	0x154F	1A9E

6.2.3 DRP

Variables	MCU Address	DSP Address	Offset in bytes
a_drp_ramp[20]	0xFFD0 1DB8	0x16DC	1DB8
d_drp_afc_add_api	0xFFD0 1D9E	0x16CF	1D9E
d_apcdel_bis	0xFFD0 01D0	0x08E8	01D0
d_apcdel2_bis	0xFFD0 01D2	0x08E9	01D2
d_apcdel2	0xFFD0 01D4	0x08EA	01D4

6.2.4 Background Task

Variables	MCU Address	DSP Address	Offset in bytes
d_background_enable	0xFFD0 0314	0x098A	0314
d_background_abort	0xFFD0 0316	0x098B	0316
d_background_state	0xFFD0 0318	0x098C	0318
d_max_background	0xFFD0 031A	0x098D	031A
a_background_tasks	0xFFD0 031C	0x098E	031C
a_back_task_io	0xFFD0 033C	0x099E	033C

6.2.5 Dynamic download

Variables	MCU Address	DSP Address	Offset in bytes
d_api_dwl_download_ctrl	0xFFD0 1FEC	0x17F6	1FEC
d_api_dwl_error_code	0xFFD0 1FEE	0x17F7	1FEE
d_api_dwl_function_address	0xFFD0 1FF0	0x17F8	1FF0
d_api_dwl_crc	0xFFD0 1FF4	0x17FA	1FF4
d_api_dwl_size	0xFFD0 1FF6	0x17FB	1FF6
d_api_dwl_write_pointer	0xFFD0 1FF8	0x17FC	1FF8

6.2.6 DSP Trace

Variables	MCU Address	DSP Address	Offset in bytes
p_debug_buffer	0xFFD0 01BE	0x08DF	01BE
d_debug_buffer_size	0xFFD0 01C0	0x08E0	01C0
d_debug_trace_type	0xFFD0 01C2	0x08E1	01C2

6.2.7 C-Port

Variables	MCU Address	DSP Address	Offset in bytes
d_cport_init	0xFFD0 0984	0x0CC2	0984
d_cport_ctrl	0xFFD0 0986	0x0CC3	0986
a_cport_cfr	0xFFD0 0988	0x0CC4	0988
d_cport_tcl_tadt	0xFFD0 098C	0x0CC6	098C
d_cport_tdat	0xFFD0 098E	0x0CC7	098E
d_cport_tvs	0xFFD0 0990	0x0CC8	0990
d_cport_status	0xFFD0 0992	0x0CC9	0992
d_cport_reg_value	0xFFD0 0994	0x0CCA	0994

6.2.8 Modem

6.2.8.1 A5 1/2

6.2.8.1.1 A5 1/2 Old driver

Variables	MCU Address	DSP Address	Offset in bytes
d_a5_mode	0xFFD0 0376	0x09BB	0376
a_kc[4]	0xFFD0 0476	0x0A3B	0476

6.2.8.1.2 A5 1/2/3 New driver (not enabled)

Variables	MCU Address	DSP Address	Offset in bytes
d_a5_mode	0xFFD0 0376	0x09BB	0376
a_a5_kc[8]	0xFFD0 6918	0x3C8C	6918
a_a5_count	0xFFD0 6F8A	0x3FC5	6F8A

6.2.8.2 FB detection

Variables	MCU Address	DSP Address	Offset in bytes
d_fb_det	0xFFD0 01F0	0x08F8	01F0
d_fb_mode	0xFFD0 01F2	0x08F9	01F2
d_fb_toa/a_sync_demod	0xFFD0 01F4	0x08F9	01F4
d_fb_pm	0xFFD0 01F6	0x08FA	01F6
d_fb_angle	0xFFD0 01F8	0x08FC	01F8
d_fb_snr	0xFFD0 01FA	0x08FE	01FA

6.2.8.3 DCO

Variables	MCU Address	DSP Address	Offset in bytes
d_dco_algo_mode	0xFFD0 6928	0x3C94	6928
d_dco_samples_per_symbol	0xFFD0 692A	0x3C95	692A
a_dco_calc_addr_i[4]	0xFFD0 692C	0x3C96	692C
a_dco_calc_addr_q[4]	0xFFD0 6934	0x3C9A	6934
d_dco_num_sample_nb	0xFFD0 693C	0x3C9E	693C
d_dco_num_sample_sb	0xFFD0 693E	0x3C9F	693E
d_dco_num_sample_pw	0xFFD0 6940	0x3CA0	6940
d_dco_csf_attenuation	0xFFD0 6942	0x3CA1	6942
d_dco_fcw	0xFFD0 6944	0x3CA2	6944
d_dco_starting_offset_int	0xFFD0 6946	0x3CA3	6946
d_dco_starting_offset_fraction	0xFFD0 6948	0x3CA4	6948

6.2.8.4 SAIC

Variables	MCU Address	DSP Address	Offset in bytes
d_swh_SaicFlag	0xFFD068AC	0x3C56	68AC

6.2.9 Acoustic Improvements

6.2.9.1 FIR uplink

Variables	MCU Address	DSP Address	Offset in bytes
d_audio_gain_ul	0xFFD0 0206	0x0903	0206
coefs_uplink/FIR31_uplink	0xFFD068AC	0x3C56	0908

6.2.9.2 IIR module

Variables	MCU Address	DSP Address	Offset in bytes
d_iir_input_scaling	0xFFD0 1AA0	0x1550	1AA0
d_iir_fir_scaling	0xFFD0 1AA2	0x1551	1AA2
d_iir_input_gain_scaling	0xFFD0 1AA4	0x1552	1AA4
d_iir_output_gain_scaling	0xFFD0 1AA6	0x1553	1AA6
d_iir_output_gain	0xFFD0 1AA8	0x1554	1AA8
d_iir_feedback	0xFFD0 1AAA	0x1555	1AAA
d_iir_nb_iir_blocks	0xFFD0 1AAC	0x1556	1AAC
d_iir_nb_fir_blocks	0xFFD0 1AAE	0x1557	1AAE
a_iir_iir_coefs	0xFFD0 1AB0	0x1558	1AB0
a_iir_fir_coefs	0xFFD0 1B50	0x15A8	1B50

6.2.9.3 Limiter module

Variables	MCU Address	DSP Address	Offset in bytes
a_lim_mul_low	0xFFD0 1BAE	0x15D7	1BAE
a_lim_mul_high	0xFFD0 1BB2	0x15D9	1BB2
d_lim_gain_fall_q15	0xFFD0 1BB6	0x15DB	1BB6
d_lim_gain_rise_q15	0xFFD0 1BB8	0x15DC	1BB8
d_lim_block_size	0xFFD0 1BBA	0x15DD	1BBA
d_lim_nb_fir_coefs	0xFFD0 1BBC	0x15DE	1BBC
d_lim_slope_update_period	0xFFD0 1BBE	0x15DF	1BBE
a_lim_filter_coefs	0xFFD0 1BC0	0x15E0	1BC0

6.2.10 Audio Applications

6.2.10.1 Tones/VoiceMemo/Melody/keybeeps

Variables	MCU Address	DSP Address	Offset in bytes
d_audio_init	0xFFD0 020C	0x0906	020C
d_audio_status	0xFFD0 020E	0x0907	020E
d_toneskb_init	0xFFD0 0210	0x0908	0210
d_toneskb_status	0xFFD0 0212	0x0909	0212
d_k_x1_t0	0xFFD0 0214	0x090A	0214
d_k_x1_t1	0xFFD0 0216	0x090B	0216
d_k_x1_t2	0xFFD0 0218	0x090C	0218
d_pe_rep	0xFFD0 021A	0x090D	021A
d_pe_off	0xFFD0 021C	0x090E	021C
d_se_off	0xFFD0 021E	0x090F	021E
d_bu_off	0xFFD0 0220	0x0910	0220
d_t0_on	0xFFD0 0222	0x0911	0222
d_t0_off	0xFFD0 0224	0x0912	0224
d_t1_on	0xFFD0 0226	0x0913	0226
d_t1_off	0xFFD0 0228	0x0914	0228
d_t2_on	0xFFD0 022A	0x0915	022A
d_t2_off	0xFFD0 022C	0x0916	022C
d_k_x1_kt0	0xFFD0 022E	0x0917	022E
d_k_x1_kt1	0xFFD0 0230	0x0918	0230
d_dur_kb	0xFFD0 0232	0x0919	0232
d_shiftdl	0xFFD0 0234	0x091A	0234
d_shiftul	0xFFD0 0236	0x091B	0236
d_melo_ctrl	0xFFD0 023E	0x091F	023E
d_melo_max_used	0xFFD0 0240	0x0920	0240
melo 0	0xFFD0 0242	0x0921	0242
melo 1	0xFFD0 024A	0x0925	024A
melo 2	0xFFD0 0252	0x0929	0252
melo 3	0xFFD0 025A	0x092D	025A
melo 4	0xFFD0 0262	0x0931	0262
melo 5	0xFFD0 026A	0x0935	026A
melo 6	0xFFD0 0272	0x0939	0272
melo 7	0xFFD0 027A	0x093D	027A
d_melody_select	0xFFD0 0282	0x0941	0282

6.2.10.2 C-Port Test

Variables	MCU Address	DSP Address	Offset in bytes
d_cport_api_dma_install	0xFFD0 25A0	0x1AD0	25A0
d_cport_api_dma_channel	0xFFD0 25A2	0x1AD1	25A2
d_cport_api_dma_rootcause	0xFFD0 25A4	0x1AD2	25A4

6.2.10.3 MP3

Variables	MCU Address	DSP Address	Offset in bytes
d_mp3_api_header	0xFFD0 25A0	0x1AD0	25A0
d_mp3_api_channel	0xFFD0 25A4	0x1AD2	25A4
d_mp3_api_init	0xFFD0 25A6	0x1AD3	25A6
d_mp3_api_play	0xFFD0 25A8	0x1AD4	25A8
d_mp3_api_pause	0xFFD0 25AA	0x1AD5	25AA
d_mp3_api_restart	0xFFD0 25AC	0x1AD6	25AC
d_mp3_api_stop	0xFFD0 25AE	0x1AD7	25AE
d_mp3_api_end	0xFFD0 25B0	0x1AD8	25B0
d_mp3_api_request_index	0xFFD0 25B2	0x1AD9	25B2
d_mp3_api_request_size	0xFFD0 25B4	0x1ADA	25B4
d_mp3_api_provided_size	0xFFD0 25B6	0x1ADB	25B6
d_mp3_api_error_code	0xFFD0 25B8	0x1ADC	25B8
a_mp3_api_input_buffer	0xFFD0 25C0	0x1AE0	25C0
a_mp3_api_output_buffer	0xFFD0 2C00	0x1E00	2C00

6.2.10.4 TTY

Variables	MCU Address	DSP Address	Offset in bytes
d_tty_status	0xFFD0 04FE	0x0A7F	04FE
d_tty_detec_thres	0xFFD0 0500	0x0A80	0500
d_ctm_detec_shift	0xFFD0 0502	0x0A81	0502
d_tty_fa_thres	0xFFD0 0504	0x0A82	0504
d_tty_mod_norm	0xFFD0 0506	0x0A83	0506
d_tty_reset_buffer_ul	0xFFD0 0508	0x0A84	0508
d_tty_loop_ctrl	0xFFD0 050A	0x0A85	050A
p_tty_loop_buffer	0xFFD0 050C	0x0A86	050C
a_TTY_FIFO_1	0xFFD0 1C0A	0x1605	1C0A
a_TTY_FIFO_2	0xFFD0 1C4C	0x1626	1C4C
a_TTY_FIFO_3	0xFFD0 1CCE	0x1667	1CCE
a_TTY_FIFO_4	0xFFD0 1D50	0x16A8	1D50

6.3 Address list in API – GSM only

6.3.1 Control

6.3.1.1 AMR

Variables	MCU Address	DSP Address	Offset in bytes
d_amrschd_api_amr_1	0xFFD0 19CA	0x14E5	19CA
d_amrschd_api_amr_2	0xFFD0 19CC	0x14E6	19CC
d_amrschd_api_amr_3	0xFFD0 19CE	0x14E7	19CE
d_amrschd_api_amr_4	0xFFD0 19D0	0x14E8	19D0
a_amrschd_raatsch_ul[6]	0xFFD0 19D2	0x14E9	19D2
a_amrschd_raatsch_dl[6]	0xFFD0 19DE	0x14EF	19DE
d_amrschd_snr_est	0xFFD0 19EA	0x14F5	19EA
d_amrschd_thr_onset_afs	0xFFD0 19EE	0x14F7	19EE
d_amrschd_thr_sid_first_afs	0xFFD0 19F0	0x14F8	19F0
d_amrschd_thr_ratscch_afs	0xFFD0 19F2	0x14F9	19F2
d_amrschd_thr_update_afs	0xFFD0 19F4	0x14FA	19F4
d_amrschd_thr_onset_ahs	0xFFD0 19F6	0x14FB	19F6
d_amrschd_thr_sid_ahs	0xFFD0 19F8	0x14FC	19F8
d_amrschd_thr_ratscch_marker	0xFFD0 19FA	0x14FD	19FA
d_amrschd_thr_snr_no_data	0xFFD0 19FC	0x14FE	19FC
d_thr_sp_dgr	0xFFD0 19FC	0x14FE	19FC
d_amrschd_thr_soft_bits	0xFFD0 19FE	0x14FF	19FE
a_d_macc_thr_afs[8]	0xFFD0 1A3C	0x151E	1A3C
a_d_macc_thr_ahs[6]	0xFFD0 1A4C	0x1526	1A4C

6.3.2 Speech Acoustics

6.3.2.1 AEC

Variables	MCU Address	DSP Address	Offset in bytes
d_ctrl_aec	0xFFD0 0238	0x091C	238
d_es_level_api	0xFFD0 023A	0x091D	23A
d_mu_api	0xFFD0 023C	0x091E	23C
d_cont_filter_api	FFD0084A	C25	84A
d_granularity_att_api	FFD0084C	C26	84C
d_coef_smooth_api	FFD0084E	C27	84E
d_es_level_max_api	FFD00850	C28	850
d_fact_vad_api	FFD00852	C29	852
d_thr_abs_api	FFD00854	C2A	854
d_fact_asd_fil_api	FFD00856	C2B	856
d_fact_asd_mut_api	FFD00858	C2C	858
d_far_end_pow_h_api	FFD0085A	C2D	85A
d_far_end_pow_l_api	FFD0085C	C2E	85C
d_far_end_noise_h_api	FFD0085E	C2F	85E
d_far_end_noise_l_api	FFD00860	C30	860

6.3.2.2 ANR

Variables	MCU Address	DSP Address	Offset in bytes
d_cntr_anr_min_gain	0xFFD0 1B90	0x15C8	1B90
d_cntr_anr_vad_thr	0xFFD0 1B92	0x15C9	1B92
d_cntr_anr_gamma_slow	0xFFD0 1B94	0x15CA	1B94
d_cntr_anr_gamma_fast	0xFFD0 1B96	0x15CB	1B96
d_cntr_anr_gamma_gain_slow	0xFFD0 1B98	0x15CC	1B98
d_cntr_anr_gamma_gain_fast	0xFFD0 1B9A	0x15CD	1B9A
d_cntr_anr_thr2	0xFFD0 1B9C	0x15CE	1B9C
d_cntr_anr_thr4	0xFFD0 1B9E	0x15CF	1B9E
d_cntr_anr_thr5	0xFFD0 1BA0	0x15D0	1BA0
d_cntr_anr_mean_ratio_thr1	0xFFD0 1BA2	0x15D1	1BA2
d_cntr_anr_mean_ratio_thr2	0xFFD0 1BA4	0x15D2	1BA4
d_cntr_anr_mean_ratio_thr3	0xFFD0 1BA6	0x15D3	1BA6
d_cntr_anr_mean_ratio_thr4	0xFFD0 1BA8	0x15D4	1BA8
d_cntr_anr_div_factor_shift	0xFFD0 1BAA	0x15D5	1BAA
d_cntr_anr_ns_level	0xFFD0 1BAC	0x15D6	1BAC

6.3.2.3 Echo suppressor

Variables	MCU Address	DSP Address	Offset in bytes
d_es_mode	0xFFD0 1BE0	0x15F0	1BE0
es_gain_dl	0xFFD0 1BE2	0x15F1	1BE2
es_gain_ul_1	0xFFD0 1BE4	0x15F2	1BE4
es_gain_ul_2	0xFFD0 1BE6	0x15F3	1BE6
d_es_tcl_fe_ls_thr	0xFFD0 1BE8	0x15F4	1BE8
d_es_tcl_dt_ls_thr	0xFFD0 1BEA	0x15F5	1BEA
d_es_tcl_fe_ns_thr	0xFFD0 1BEC	0x15F6	1BEC
d_es_tcl_dt_ns_thr	0xFFD0 1BEE	0x15F7	1BEE
d_es_tcl_ne_thr	0xFFD0 1BF0	0x15F8	1BF0
d_es_ref_ls_pwr	0xFFD0 1BF2	0x15F9	1BF2
d_es_switching_time	0xFFD0 1BF4	0x15FA	1BF4
d_es_switching_time_dt	0xFFD0 1BF6	0x15FB	1BF6
d_es_hang_time	0xFFD0 1BF8	0x15FC	1BF8
a_es_gain_lin_dl_vect[0]	0xFFD0 1BFA	0x15FD	1BFA
a_es_gain_lin_dl_vect[1]	0xFFD0 1BFC	0x15FE	1BFC
a_es_gain_lin_dl_vect[2]	0xFFD0 1BFE	0x15FF	1BFE
a_es_gain_lin_dl_vect[3]	0xFFD0 1C00	0x1600	1C00
a_es_gain_lin_ul_vect[0]	0xFFD0 1C02	0x1601	1C02
a_es_gain_lin_ul_vect[1]	0xFFD0 1C04	0x1602	1C04
a_es_gain_lin_ul_vect[2]	0xFFD0 1C06	0x1603	1C06
a_es_gain_lin_ul_vect[3]	0xFFD0 1C08	0x1604	1C08

6.4 Address list in API – NDB GPRS only

Variables	MCU Address	DSP Address	Offset in bytes
d_sched_mode_gprs	0xFFD0 0378	0x09BC	0378
d_usf_updated_gprs	0xFFD0 0384	0x09C2	0384
d_win_start_gprs	0xFFD0 0386	0x09C3	386
d_usf_vote_enable	0xFFD0 0388	0x09C4	0388
a_ctrl_ched_gprs	0xFFD0 0390	0x09C8	0390
a_ul_buffer_gprs	0xFFD0 03A0	0x09D0	03A0
a_usf_gprs	0xFFD0 03B0	0x09D8	03B0
a_interference_meas_gprs	0xFFD0 03C0	0x09E0	03C0
a_ptcchu_gprs	0xFFD0 03D0	0x09E8	03D0
a_dd_md_gprs	0xFFD0 03D8	0x09EC	03D8
a_du_gprs[0]	0xFFD0 03F8	0x09FC	03F8
a_du_gprs[1]			0432
a_du_gprs[2]			046C
a_du_gprs[3]			04A6
a_pu_gprs[0]	0xFFD0 04E0	0x0A70	04E0
a_pu_gprs[1]			04FE
a_pu_gprs[2]			051C
a_pu_gprs[3]			053A
d_rlcmac_rx_no_gprs	FFD00558	AAC	558
a_dd_gprs[0]	FFD0055A	AAD	55A
a_dd_gprs[1]			0598
a_dd_gprs[2]			05D6
a_dd_gprs[3]			0614
a_dd_gprs[4]			0652
a_dd_gprs[5]			0690
a_dd_gprs[6]			06CE
a_dd_gprs[7]			070C
a_drp_ramp2_gprs[0]	0xFFD0 1E00	0x1700	1E00
a_drp_ramp2_gprs[1]	0xFFD0 1E28	0x1714	1E28
a_drp_ramp2_gprs[2]	0xFFD0 1E50	0x1728	1E50
a_drp_ramp2_gprs[3]	0xFFD0 1E78	0x173C	1E78
a_drp_ramp2_gprs[4]	0xFFD0 1EA0	0x1750	1EA0
a_drp_ramp2_gprs[5]	0xFFD0 1EC8	0x1764	1EC8
a_drp_ramp2_gprs[6]	0xFFD0 1EF0	0x1778	1EF0
a_drp_ramp2_gprs[7]	0xFFD0 1F18	0x178C	1F18

7 Parameters address list

7.1 Common GSM and GPRS Parameter address List

Variables	MCU Address	DSP Address	Offset in bytes
d_transfer_rate	0xFFD0 0862	0x0C31	0862
d_lat_mcu_bridge	0xFFD0 0864	0x0C32	0864
d_lat_mcu_hom2sam	0xFFD0 0866	0x0C33	0866
d_lat_mcu_bef_fast_access	0xFFD0 0868	0x0C34	0868
d_lat_dsp_after_sam	0xFFD0 086A	0x0C35	086A
p_gprs_install_address	0xFFD0 086C	0x0C36	086C
d_dmacc_threshold	0xFFD0 0902	0x0C81	0902
d_sd_threshold	0xFFD0 0904	0x0C82	0904
d_nb_max_iterations	0xFFD0 0906	0x0C83	0906
d_gsm_bgd_mgt	0xFFD0 08FE	0x0C7F	08FE
d_cn_sw_workaround	0xFFD0 0870	0x0C38	0870

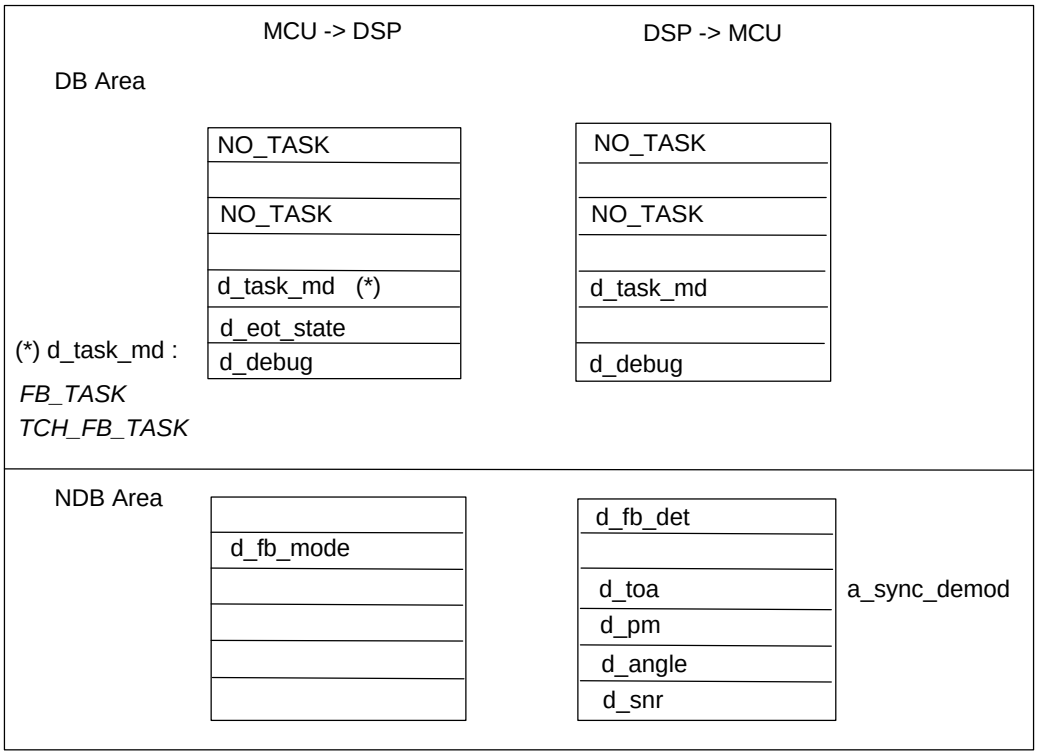
7.2 GPRS Only Parameter address List

Variables	MCU Address	DSP Address	Offset in bytes
d_thr_usf_detect	0xFFD0 0374	0x09BA	0374

8 DETAILS ON THE INTERFACE – Common to GSM and GPRS

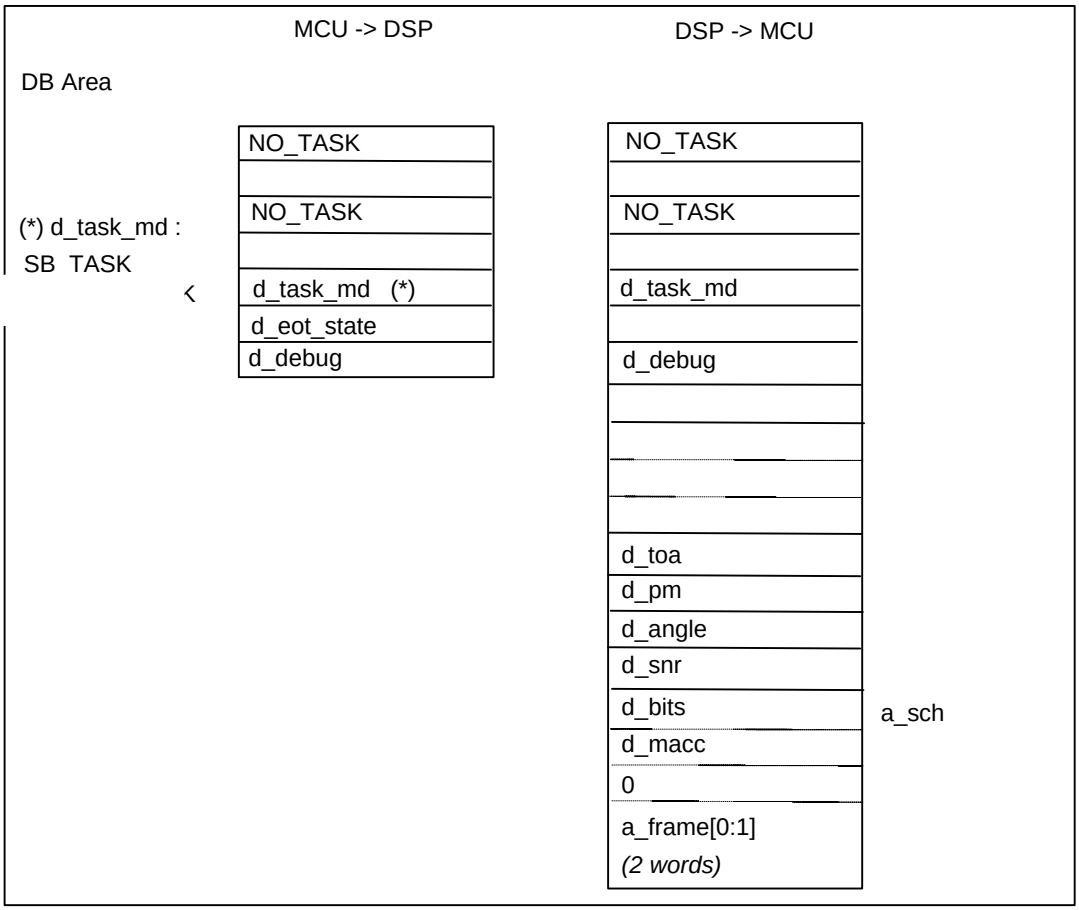
8.1 Frequency Burst Tasks

Functional Description : DSP performs frequency burst detection in communication window.
Communication Period : In Idle Mode, around 12 "TDMAs" ((Ctrl-12Wrk-Rd))
In Traffic Channel Mode, around 9 timeslots
Number of periods : In Idle Mode, 1 per frequency burst detection attempt
In Traffic Channel, 10 per frequency burst detection attempt



8.2 Synchro Burst Tasks

Functional Description : DSP performs synchro burst decoding (or part of it)
Communication Period : - In Idle mode 2 “TDMAs” (Ctrl-2Wrk-Rd)
- In Traffic Channel : 9 timeslots
Number of periods : In Idle mode : 2 per attempt



8.3 Description of the Tone/Keybeep/Memo Tasks (MISC Task)

Each type of tasks is described by a set of parameters and trigger/status flags .

d_toneskb_init :

Bit0: Set to 1 when MCU wants to start tone generation. This bit will be cleared by DSP when initialization is acknowledged.
Bit1: Set to 1 when MCU wants to start keybeep generation. This bit will be cleared by DSP when initialization is acknowledged.
Bit2: Set to 1 when MCU wants to start recording message. Cleared by DSP when initialization is acknowledged.
Bit3: Set to 1 when MCU wants to stop recording message. Cleared by DSP when initialization is acknowledged.
Bit4: Set to 1 when MCU wants to start playing message. Cleared by DSP when initialization is acknowledged.
Bit5: Set to 1 when MCU wants to stop playing message. Cleared by DSP when initialization is acknowledged.
Bit6: Set to 1 when MCU wants to play tones in downlink & in uplink. Cleared by DSP when tone is ended.
Bit7: Unused
Bit8: Unused
Bit9: Unused
Bit10: Unused
Bit11: Unused
Bit12: Unused
Bit13: Unused.
Bit14: Unused
Bit15: Unused

Prior tone or keybeep is started (Bit0 or Bit1 set to 1) , corresponding parameters must be loaded.

d_toneskb_status:

Bit0: Set to 1 when DSP starts tone generation. Cleared by DSP when tone has been generated.
Bit1: Set to 1 when DSP starts keybeep generation. Cleared by DSP when keybeep has been generated.
Bit2: Set to 1 when DSP starts recording speech for voice memo / MMS-dictaphone purpose. Cleared by DSP when operation has been processed.
Bit3: Set to 1 when DSP starts playing speech for voice memo / MMS-dictaphone purpose. Cleared by DSP when operation has been processed
Bit4: Unused
Bit5: Unused
Bit6: Unused
Bit7: Unused
Bit8: Unused
Bit9: Set to 1 to request next MISC TDMA interrupt to finish the downlink audio samples transmission at the end of processing.
Bit10: Unused
Bit11: Unused
Bit12: Unused
Bit13: Set to 1 when the DSP starts the melody generator module. Set to 0 when the DSP stops the module.
Bit14: Unused
Bit15: Unused

Independently from each other, the DSP is able to generate two types of tones:
- signaling tones
- key beep tones.

The Tone/Keybeep task is an msc task that can be performed concurrently to any gsm task. Any concurrent ongoing tone or keybeep will be interrupted during transition to TCH speech modes or signaling only mode or at voice coder resynchronization (handover).

MCU needs to activate DSP in MSC task mode at each communication interrupt during tone/keybeep generation. d_toneskb_status allows MCU to stop enabling communication interrupt in MSC mode as soon as tone/keybeep has been generated.

When MCU wants to stop ongoing tone or keybeep, this can be done programming NULL tone or NULL keybeep. NULL tone is corresponding to tone which period is 0 (d_pe_rep=0) or keybeep which duration is 0 (d_dur_kb=0).

8.3.1.1 Tri-tones generator algorithm

Possible signaling tone signals consists of:
- a burst which consists of 1 up to 3 tones (alternately or in parallel) beginning and end of each tone is independently programmable.
- a sequence which consists of a multiple of bursts. Bursts are automatically repeated within a sequence
- a period which consists of a sequence and, if required, a silent phase. The period is repeated according to a programmable period count as a special case, the sequence may be repeated indefinitely.

8.3.1.2 tri-tones Parameters

- d_k_x1_t*X* *X*[0..2]
- d_pe_rep
- d_pe_off
- d_se_off
- d_bu_off
- d_t*X*_on *X*[0..2]
- d_t*X*_off *X*[0..2]

- **d_k_x1_t*X*** *X*[0..2]: this variable is a 16 bits word and describes the frequency (8 most significant bits) and the amplitude (8 less significant bits) of tone X.

0 < f < 2kHz, fs = 8kHz :
The coefficient k_tX = d_k_x1_tX[8..15] determines the frequency :

$$k\_tX = \cos(2 \pi f/fs)$$

The coefficient x1_tX = d_k_x1_tX[0..7] determines the amplitude :

$$x1\_tX = amp * \sin(2 \pi f/fs)$$

Since the amplitude value **amp** has to be multiplied with the correction factor (sin(2 π f/fs)), it depends on the frequency.
So when one wants to create a tone with a certain attenuation to the maximum amplitude, he first has to compute the value of **amp** from :

$$amplitude(dB) = 20 \log10 (amp) - 48.16$$

In fact, **amp** value for a given ***amplitude*** can be deduce from Table 0.2, and then multiply it with the correction factor given in Table 0.1.

For example :
If we want to process tone 0 with **f = 900 Hz** and **amplitude = -20.56 dB :**
We will find in Table 1, **k_t0 = 195** (fixed point arithmetic value **F0.8**). In fact, in floating point arithmetic : k_t0 = cos (2 π 900 / 8000) = 0.76.
In order to have the same accuracy in fixed point arithmetic (2 digits), we will need at least 7 bits. Furthermore, we will always deal with positive value lower than 1 so we won't need any bit to code the entire part of k_t0.
In conclusion, in fixed point arithmetic, we will then use the 8 bits at our disposal to code the fractional part of k_t0 (**F0.8**).

We will find in Table 2, **amp = 24** and in Table 1 **correction_factor(900) = 0.648 :**
x1_t0 = amp * correction_factor = 24 * 0.648 = 15.552 (floating point value)
0<amp<256 and 0<correction_factor<1, so 0<amp*correction_factor<256. We will need the 8 bits at our disposal to code the entire part of x1_t0 (**F8.0**).
x1_t0 = 15 (fixed point arithmetic value, **F8.0**)

- **d_pe_rep** : number of periods to process. ***0<=d_pe_rep<=0xffff***.

If ***d_pe_rep=0***, we are running a 'NULL' tone. Nothing occurs except switching b_tg_st to 0.
If ***d_pe_rep=0xffff***, it means that we are running tones indefinitely. In this case, the process can only be stopped by sending a 'NULL' tone.

- **d_pe_off** : duration of a period. A period consists of a sequence and, if required, a silent phase. This duration corresponds to a number of audio frames.

0 <= d_pe_off <= 32767 (1=20 ms)

- **d_se_off** : duration of a sequence. This duration corresponds to a number of audio frames. ***0 <= d_se_off <= d_pe_off. d_se_off = n . d_bu_off***

- **d_bu_off** : duration of a burst. This duration corresponds to a number of audio frames.

0 <= d_bu_off<=d_se_off<=d_pe_off.

- **d_t*X*_on** *X*[0..2] : starting time of tone *X*. Considering the length of a burst period, in term of number of audio frames, this value describes when the considered tone must be started (audio_frame[0], audio_frame[1]…audio_frame[d_bu_off-1]).

0 <= d_tX_on <= d_bu_off

- **d_t*X*_off** *X*[0..2] : ending time of tone *X*. Considering the length of a burst period, in term of number of audio frames, this value describes when the considered tone must be stopped (audio_frame[0], audio_frame[1]…audio_frame[d_bu_off - 1]).

0 <= d_tX_on <= d_tX_off

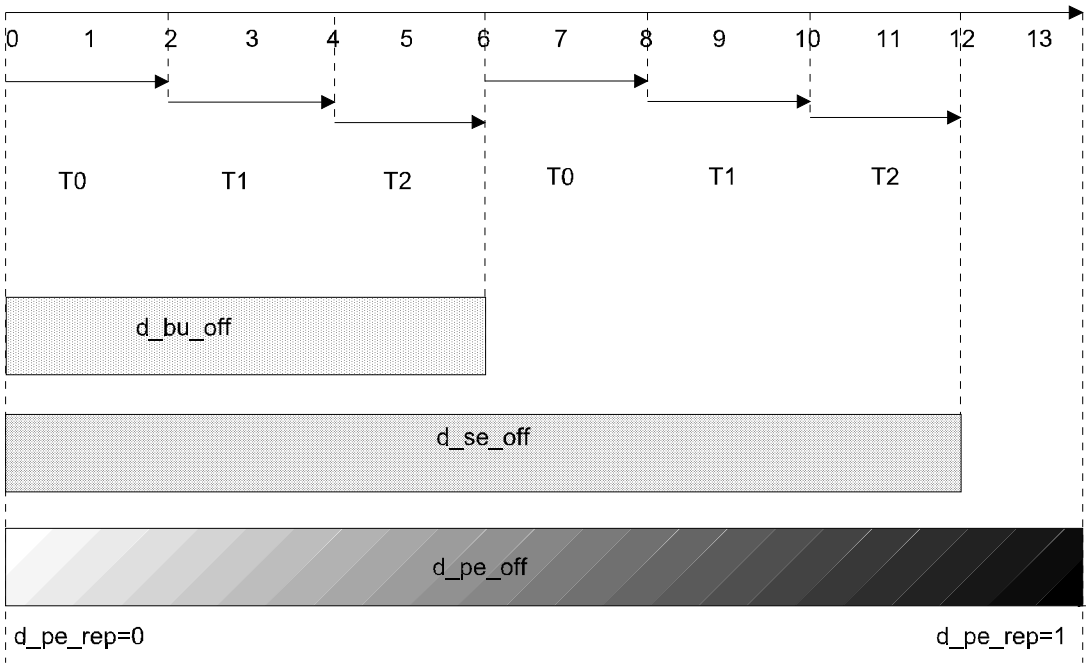


Figure 1. Detailed example of timing parameter

In the previous example:

d_pe_rep	: 1	
d_pe_off	: 13	260ms
d_se_off	: 12	240ms
d_bu_off	: 6	120ms
d_t0_on, d_t0_off	: 0 , 2	0, 40ms
d_t1_on, d_t1_off	: 2 , 4	40, 80ms
d_t2_on, d_t2_off	: 4 , 6	80, 120ms

8.3.1.3 key beeps Parameters

- d_k_x1_kt*X* *X*[0..1]
- d_dur_kb

- **d_k_x1_kt*X*** *X*[0..1] : This variable is a 16 bits word and describes the frequency (8 most significant bits) and the amplitude (8 less significant bits) of tone X. The coefficient k_tX = d_k_x1_tX[8..15] determines the frequency. **k_tX = cos(2** p **f/fs)** 0<f<2 kHz. The coefficient x1_tX = d_k_x1_tX[0..7] determines the amplitude. **x1_tX=amplitude*sin(2** p **f/fs).**

The way to set k_tX and x1_tX is exactly the same than the one presented for d_k_c1_tX variable.

- **d_dur_kb** : duration of key beep. This duration corresponds to a number of audio frames.

0 <= d_dur_kb <= 32767 (1=20 ms)

9 DETAILS ON THE INTERFACE – GSM only

9.1 Buffers formats

9.1.1 Format for TCH/FS

a_dd_0, a_dd_1, a_du_0, a_du_1 buffers for TCH modes

a_dd_0 and a_dd_1 buffers are used for downlink transfers.

- a_dd_0: downlink information from network
- a_dd_1: play information from MCU. This mode is used mainly for tests. In this mode, the information to voice decoder is taken from a_dd_1 (MCU updating buffer). This mode is selected by setting *b_play_dl* to 1 in *d_tch_mode* register.

a_du_0 and a_du_1 buffers are used for uplink transfers.

- a_du_0: uplink information to network for speech and datas. For speech, information is taken from encoder.
- a_du_1: play information from MCU. This mode is used for tests. In this mode, the information to channel encoder is taken from a_du_1 (MCU updating buffer). This mode is selected by setting *b_play_ul* to 1 in *d_tch_mode* register.

Use : a_du_0 (Uplink), a_dd_0 (Downlink)

Uplink :
- d-bits :
  - b_blud : “1” speech frame to be transmitted available, “0” no frame available
    (*) Channel encoder copies this flag to the t-flags in the interleaver.
  - b_af : speech activity flag computed by speech encoder VAD-function.
  - b_efrcrc, b_sid0, b_sid1, b_bfi, b_bci, b_ufi : not used in encoder.
- d_macc : not used in encoder.
- d_nerr : not used in encoder.

Downlink :
- d-bits :
  - b_blud : set to 1 by decoder as default.
  - b_af : set to 1 by decoder as default.
  - b_efrcrc : set to 1 by decoder as default.
  - b_sid1, b_sid0 : silence identification bits
    (sid1,sid0) : (0, 0) no SID
    (0, 1) invalid SID
    (1, 0) valid SID
  - b_bfi : bad frame flag, set by channel decoder.
  - b_bci, b_ufi : not used, set to 0 by decoder as default.
- d_macc : accumulated metric (max. 21168 = 0x52b0)
- d_nerr : number of estimated errors (max. 378 = 0x017a)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	blud	af	0	0	0	0	efr crc	0	0	0	0	sid1	sid0	bfi	bci	ufi
1	d_macc															
2	d_nerr															
3	c1 (0)															c1 (15)
4	(16)															(31)
5	(32)															(47)
6	(48)															(63)
7	(64)															(79)
8	(80)															(95)
9	(96)															(111)
10	(112)															(127)
11	(128)															(143)
12	(144)															(159)
13	(160)															(175)
14	c1 (176)					c1 (181)	0	0	0	0	c2 (0)					c2 (5)
15	(6)															(21)
16	(22)															(37)
17	(38)															(53)
18	(54)															(69)
19	c2 (70)							c2 (77)	0	0	0	0	0	0	0	0

9.1.2 Format for TCH/EFS

Use : a_du_0 (Uplink), a_dd_0 (Downlink)

Uplink :

d-bits :

b_blud : “1” speech frame to be transmitted available, “0” no frame available

(*) Channel encoder copies this flag to the t-flags in the interleaver.

b_af : speech activity flag computed by speech encoder VAD-function.

b_efrcrc, b_sid0, b_sid1, b_bfi, b_bci, b_ufi : not used in encoder.

d_macc : not used in encoder.

d_nerr : not used in encoder.

Downlink :

d-bits :

b_blud : set to 1 by decoder as default.

b_af : set to 1 by decoder as default.

b_efrcrc : “1” error in EFR 8bits CRC, “0” no error in EFR CRC.

b_sid1, b_sid0 : silence identification bits.

(sid1,sid0) : (0, 0) no SID

(0, 1) invalid SID

(1, 0) valid SID

b_bfi : bad frame flag, set by channel decoder.

b_bci, b_ufi : not used, set to 0 by decoder as default.

d_macc : accumulated metric (max. 21168 = 0x52b0)

d_nerr : number of estimated errors (max. 378 = 0x017a)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	blud	af	0	0	0	0	efr crc	0	0	0	0	sid1	sid0	bfi	bci	ufi
1	d_macc															
2	d_nerr															
3	c1 (0)															c1 (15)
4	(16)															(31)
5	(32)															(47)
6	(48)															(63)
7	c1 (64)	0 (*)	0 (*)	0 (*)	0 (*)	0 (*)	0 (*)	0 (*)	0 (*)	c1 (73)						c1 (79)
8	(80)															(95)
9	(96)															(111)
10	(112)															(127)
11	(128)															(143)
12	(144)															(159)
13	(160)															(175)
14	c1 (176)					c1 (181)	0	0	0	0	c2 (0)	c2 (**)	c2 (**)	c2 (3)	c2 (**)	c2 (**)
15	c2 (6)	(**)	(**)	c2 (9)	(**)	(**)	c2 (12)									c2 (21)
16	(22)															(37)
17	(38)															(53)
18	(54)															(69)
19	c2 (70)							c2 (77)	0	0	0	0	0	0	0	0

(*) EFR 8 bits CRC position. Must be all zero for encoder input. Decoder clears these positions.

(**) EFR repetition bits position. These bits must be encoded before channel encoder, i.e. in subjective encoder. Channel decoder takes decisions for these bits and puts results in the buffer.

9.1.3 Format for TCH/HS

Use : a_du_0, a_du_1 (Uplink), a_dd_0, a_dd_1 (Downlink)

Uplink :

d-bits :

b_blud : “1” speech frame to be transmitted available, “0” no frame available
(*) Channel encoder copies this flag to the t-flags in the interleaver.
b_af : speech activity flag computed by speech encoder VAD-function.
b_sid0, b_sid1, b_bfi, b_bci, b_ufi : not used in encoder.

d_macc : not used in encoder.
d_nerr : not used in encoder.

Downlink :

d-bits :

b_blud : set to 1 by decoder as default.
b_af : set to 1 by decoder as default.
b_sid1, b_sid0 : silence identification bits.
(sid1,sid0) : (0, 0) no SID
(0, 1) invalid SID
(1, 0) valid SID
b_bfi, b_bci, b_ufi : set by channel decoder.

d_macc : accumulated metric (max. 11816 = 0x2e28)
d_nerr : number of estimated errors (max. 211 = 0x00d3)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	blud	af	0	0	0	0	0	0	0	0	0	sid1	sid0	bfi	bci	ufi
1	d_macc															
2	d_nerr															
3	c1 (0)															c1 (15)
4	(16)															(31)
5	(32)															(47)
6	(48)															(63)
7	(64)															(79)
8	c1 (80)														c1 (94)	0
9	0	0	0	c2 (0)												c2 (12)
10	c2 (13)			c2 (16)	0	0	0	0	0	0	0	0	0	0	0	0
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

“-“ : undefined.

9.1.4 Format for TCH/AFS12.2

Additional notations for all AMR channels:

b_amr_rx_type : three info bits {rxtyp [2], rxtyp [1], rxtyp [0]}
This code corresponds to the fields RX_TYPE described in the GSM 06.93
b_amr_channel_type : three info bits {amrtyp [2], amrtyp [1], amrtyp [0]}

Binary code	b_amr_channel_type	b_amr_rx_type
000	4.75	SPEECH_GOOD
001	5.15	SPEECH_DEGRADED
010	5.9	ONSET
011	6.7	SPEECH_BAD
100	7.4	SID_FIRST
101	7.95	SID_UPDATE/ RATSCCH_GOOD
110	10.2	SID_BAD/ RATSCCH_BAD
111	12.2	NO_DATA

Use : a_du_0 (Uplink), a_dd_0 (Downlink)
Uplink :
d-bits :
b_blud : flag for block transmission.
typ(i) : b_amr_channel_type = AFS_12.2 (111)
rx(i) : unused
d_met : not used in encoder.
d_nerr : not used in encoder.
Downlink :
d-bits :
b_blud : block receive flag, set to 1 by AMR scheduler as default.
typ(i) : b_amr_channel_type = AFS_12.2 (111)
rx(i) : b_amr_rx_type depending on the channel conditions : SPEECH_GOOD(0), SPEECH_DEGRADED(1), ONSET(2), SPEECH_BAD(3), SID_FIRST(4), SU_GOOD(5), SU_BAD(6), NO_DATA(7)
d_met : decoding metric (max. 32767 = 0x7fff)
d_nerr : number of estimated errors (max. 448 = 0x01c0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	blud	0	0	0	0	0	0	0	0	0	rx2	rx1	rx0	typ2	typ1	typ0
1	d_met															
2	d_nerr															
3	c1a (0)															c1a (15)
4	(16)															(31)
5	(32)															(47)
6	(48)															(63)
7	(64)															(79)
8	c1a (80)	c1b (0)														c1b (14)
9	c1b (15)															c1b (30)
10	(31)															(46)
11	(47)															(62)
12	(63)															(78)
13	(79)															(94)
14	(95)															(110)
15	(111)															(126)
16	(127)															(142)
17	(143)															(158)
18	(159)	(160)	(161)	(162)	0	0	0	0	0	0	0	0	0	0	0	0
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

"-" : undefined.

9.1.5 Format for TCH/AFS10.2

Use : a_du_0 (Uplink), a_dd_0 (Downlink)

Uplink :

d-bits :

b_blud : flag for block transmission.
typ(i) : b_amr_channel_type = AFS_10.2 (110)
rx(i) : unused

d_met : not used in encoder.
d_nerr : not used in encoder.

Downlink :

d-bits :

b_blud : block receive flag, set to 1 by AMR scheduler as default.
typ(i) : b_amr_channel_type = AFS_10.2 (110)
rx(i) : b_amr_rx_type depending on the channel conditions : SPEECH_GOOD(0), SPEECH_DEGRADED(1), ONSET(2), SPEECH_BAD(3), SID_FIRST(4), SU_GOOD(5), SU_BAD(6), NO_DATA(7)

d_met : decoding metric (max. 32767 = 0x7fff)
d_nerr : number of estimated errors (max. 448 = 0x01c0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	blud	0	0	0	0	0	0	0	0	0	rx2	rx1	rx0	typ2	typ1	typ0
1	d_met															
2	d_nerr															
3	c1a (0)															c1a (15)
4	(16)															(31)
5	(32)															(47)
6	(48)															(63)
7	c1a (64)	c1b (0)														c1b (14)
8	c1b (15)															c1b (30)
9	(31)															(46)
10	(47)															(62)
11	(63)															(78)
12	(79)															(94)
13	(95)															(110)
14	(111)															(126)
15	(127)											(138)	0	0	0	0
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

"-" : undefined.

9.1.6 Format for TCH/AFS7.95

Use : a_du_0 (Uplink), a_dd_0 (Downlink)

Uplink:

d-bits :

b_blud : flag for block transmission.
typ(i) : b_amr_channel_type = AFS_7.95 (101)
rx(i) : unused

d_met : not used in encoder.
d_nerr : not used in encoder.

Downlink:

d-bits :

b_blud : block receive flag, set to 1 by AMR scheduler as default.
typ(i) : b_amr_channel_type = AFS_7.95 (101)
rx(i) : b_amr_rx_type depending on the channel conditions : SPEECH_GOOD(0), SPEECH_DEGRADED(1), ONSET(2), SPEECH_BAD(3), SID_FIRST(4), SU_GOOD(5), SU_BAD(6), NO_DATA(7)

d_met : decoding metric (max. 32767 = 0x7fff)
d_nerr : number of estimated errors (max. 448 = 0x01c0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	blud	0	0	0	0	0	0	0	0	0	rx2	rx1	rx0	typ2	typ1	typ0
1	d_met															
2	d_nerr															
3	c1a (0)															c1a (15)
4	(16)															(31)
5	(32)															(47)
6	(48)															(63)
7	c1a (64)	c1a (65)	c1a (66)	c1a (67)	c1a (68)	c1a (69)	c1a (70)	c1a (71)	c1a (72)	c1a (73)	c1a (74)	c1b (0)	c1b (1)	c1b (2)	c1b (3)	c1b (4)
8	(5)															(20)
9	(21)															(36)
10	(37)															(52)
11	(53)															(68)
12	c1b (69)														c1b (83)	0
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

"-" : undefined.

9.1.7 Format for TCH/AFS7.4

Use : a_du_0 (Uplink), a_dd_0 (Downlink)

Uplink :

d-bits :

b_blud : flag for block transmission.
typ(i) : b_amr_channel_type = AFS_7.4 (100)
rx(i) : unused

d_met : not used in encoder.
d_nerr : not used in encoder.

Downlink :

d-bits :

b_blud : block receive flag, set to 1 by AMR scheduler as default.
typ(i) : b_amr_channel_type = AFS_7.4 (100)
rx(i) : b_amr_rx_type depending on the channel conditions : SPEECH_GOOD(0), SPEECH_DEGRADED(1), ONSET(2), SPEECH_BAD(3), SID_FIRST(4), SU_GOOD(5), SU_BAD(6), NO_DATA(7)

d_met : decoding metric (max. 32767 = 0x7fff)
d_nerr : number of estimated errors (max. 448 = 0x01c0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	blud	0	0	0	0	0	0	0	0	0	rx2	rx1	rx0	typ2	typ1	typ0
1	d_met															
2	d_nerr															
3	c1a (0)															c1a (15)
4	(16)															(31)
5	(32)															(47)
6	(48)												c1a (60)	c1b (0)	(1)	(2)
7	(3)															(18)
8	(19)															(34)
9	(33)															(50)
10	(51)															(66)
11	(67)															(82)
12	(83)	(84)	(85)	(86)	0	0	0	0	0	0	0	0	0	0	0	0
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

"-" : undefined.

9.1.8 Format for TCH/AFS6.7

Use : a_du_0 (Uplink), a_dd_0 (Downlink)

Uplink :

d-bits :

b_blud : flag for block transmission.
typ(i) : b_amr_channel_type = AFS_6.7 (011)
rx(i) : unused

d_met : not used in encoder.
d_nerr : not used in encoder.

Downlink :

d-bits :

b_blud : block receive flag, set to 1 by AMR scheduler as default.
typ(i) : b_amr_channel_type = AFS_6.7 (011)
rx(i) : b_amr_rx_type depending on the channel conditions : SPEECH_GOOD(0), SPEECH_DEGRADED(1), ONSET(2), SPEECH_BAD(3), SID_FIRST(4), SU_GOOD(5), SU_BAD(6), NO_DATA(7)

d_met : decoding metric (max. 32767 = 0x7fff)
d_nerr : number of estimated errors (max. 448 = 0x01c0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	blud	0	0	0	0	0	0	0	0	0	rx2	rx1	rx0	typ2	typ1	typ0
1	d_met															
2	d_nerr															
3	c1a (0)															c1a (15)
4	(16)															(31)
5	(32)															(47)
6	(48)	(49)	(50)	(51)	(52)	(53)	c1a (54)	c1b (0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
7	(9)															(24)
8	(25)															(40)
9	(41)															(56)
10	(57)															(72)
11	(73)	(74)	(75)	(76)	(77)	c1b (78)	0	0	0	0	0	0	0	0	0	0
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

"-" : undefined.

9.1.9 Format for TCH/AFS5.9

Use : a_du_0 (Uplink), a_dd_0 (Downlink)

Uplink :

d-bits :

b_blud : flag for block transmission.
typ(i) : b_amr_channel_type = AFS_5.9 (010)
rx(i) : unused

d_met : not used in encoder.
d_nerr : not used in encoder.

Downlink :

d-bits :

b_blud : block receive flag, set to 1 by AMR scheduler as default.
typ(i) : b_amr_channel_type = AFS_5.9 (010)
rx(i) : b_amr_rx_type depending on the channel conditions : SPEECH_GOOD(0), SPEECH_DEGRADED(1), ONSET(2), SPEECH_BAD(3), SID_FIRST(4), SU_GOOD(5), SU_BAD(6), NO_DATA(7)

d_met : decoding metric (max. 32767 = 0x7fff)
d_nerr : number of estimated errors (max. 448 = 0x01c0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	blud	0	0	0	0	0	0	0	0	0	rx2	rx1	rx0	typ2	typ1	typ0
1	d_met															
2	d_nerr															
3	c1a (0)															c1a (15)
4	(16)															(31)
5	(32)															(47)
6	(48)	(49)	(50)	(51)	(52)	(53)	c1a (54)	c1b (0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
7	(9)															(24)
8	(25)															(40)
9	(41)															(56)
10	(57)	(58)	(59)	(60)	(61)	(62)	0	0	0	0	0	0	0	0	0	0
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

"-" : undefined.

9.1.10 Format for TCH/AFS5.15

Use : a_du_0 (Uplink), a_dd_0 (Downlink)

Uplink :

d-bits :

b_blud : flag for block transmission.
typ(i) : b_amr_channel_type = AFS_5.15 (001)
rx(i) : unused

d_met : not used in encoder.
d_nerr : not used in encoder.

Downlink :

d-bits :

b_blud : block receive flag, set to 1 by AMR scheduler as default.
typ(i) : b_amr_channel_type = AFS_5.15 (001)
rx(i) : b_amr_rx_type depending on the channel conditions : SPEECH_GOOD(0), SPEECH_DEGRADED(1), ONSET(2), SPEECH_BAD(3), SID_FIRST(4), SU_GOOD(5), SU_BAD(6), NO_DATA(7)

d_met : decoding metric (max. 32767 = 0x7fff)
d_nerr : number of estimated errors (max. 448 = 0x01c0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	blud	0	0	0	0	0	0	0	0	0	rx2	rx1	rx0	typ2	typ1	typ0
1	d_met															
2	d_nerr															
3	c1a (0)															c1a (15)
4	(16)															(31)
5	(32)															(47)
6	c1a (48)	c1b (0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)
7	(15)															(30)
8	(31)															(46)
9	(47)	(48)	(49)	(50)	(51)	(52)	(53)	0	0	0	0	0	0	0	0	0
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

"-" : undefined.

9.1.11 Format for TCH/AFS4.75

Use : a_du_0 (Uplink), a_dd_0 (Downlink)

Uplink :

d-bits :

b_blud : flag for block transmission.
typ(i) : b_amr_channel_type = AFS_4.75 (000)
rx(i) : unused

d_met : not used in encoder.
d_nerr : not used in encoder.

Downlink :

d-bits :

b_blud : block receive flag, set to 1 by AMR scheduler as default.
typ(i) : b_amr_channel_type = AFS_4.75 (000)
rx(i) : b_amr_rx_type depending on the channel conditions : SPEECH_GOOD(0), SPEECH_DEGRADED(1), ONSET(2), SPEECH_BAD(3), SID_FIRST(4), SU_GOOD(5), SU_BAD(6), NO_DATA(7)

d_met : decoding metric (max. 32767 = 0x7fff)
d_nerr : number of estimated errors (max. 448 = 0x01c0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	blud	0	0	0	0	0	0	0	0	0	rx2	rx1	rx0	typ2	typ1	typ0
1	d_met															
2	d_nerr															
3	c1a (0)															c1a (15)
4	(16)															(31)
5	(32)	(33)	(34)	(35)	(36)	(37)	c1a (38)	c1b (0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
6	(9)															(24)
7	(25)															(40)
8	(41)														(55)	0
9	-															-
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

"-" : undefined.

9.1.12 Format for TCH/AHS7.95

Use : a_du_0 or a_du_1 according to subchannel (Uplink), a_dd_0 or a_dd_1 according to subchannel(Downlink)

Uplink :

d-bits :

b_blud : flag for block transmission.
typ(i) : b_amr_channel_type = AHS_7.95 (101)
rx(i) : unused

d_met : not used in encoder.
d_nerr : not used in encoder.

Downlink :

d-bits :

b_blud : block receive flag, set to 1 by AMR scheduler as default.
prefa : FACCH has been predetected(1)
typ(i) : b_amr_channel_type = AHS_7.95 (101)
rx(i) : b_amr_rx_type depending on the channel conditions : SPEECH_GOOD(0), SPEECH_DEGRADED(1), ONSET(2), SPEECH_BAD(3), SID_FIRST(4), SU_GOOD(5), SU_BAD(6), NO_DATA(7)

d_met : decoding metric (max. 32767 = 0x7fff)
d_nerr : number of estimated errors (max. 188 = 0x00bc)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	blud	0	0	0	0	0	0	prefa	0	0	rx2	rx1	rx0	typ2	typ1	typ0
1	d_met															
2	d_nerr															
3	c1a (0)															c1a (15)
4	(16)															(31)
5	(32)															(47)
6	(48)															(63)
7	(64)	(65)	c1a (66)	c1b (0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)
8	(13)															
9	(29)															
10	(45)	(46)	(47)	(48)	(49)	(50)	(51)	(52)	(53)	(54)	c1b (55)	c2 (0)	(1)	(2)	(3)	(4)
11	(5)															(20)
12	(21)	(22)	(23)	(24)	(25)	(26)	(27)	(28)	(29)	(30)	(31)	(32)	(33)	(34)	c2 (35)	0
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

"-" : undefined.

9.1.13 Format for TCH/AHS7.4

Use : a_du_0 or a_du_1 according to subchannel (Uplink), a_dd_0 or a_dd_1 according to subchannel(Downlink)

Uplink :

d-bits :

b_blud : flag for block transmission.
typ(i) : b_amr_channel_type = AHS_7.4 (100)
rx(i) : unused

d_met : not used in encoder.
d_nerr : not used in encoder.

Downlink :

d-bits :

b_blud : block receive flag, set to 1 by AMR scheduler as default.
prefa : FACCH has been predetected(1)
typ(i) : b_amr_channel_type = AFS_7.4 (100)
rx(i) : b_amr_rx_type depending on the channel conditions : SPEECH_GOOD(0), SPEECH_DEGRADED(1), ONSET(2), SPEECH_BAD(3), SID_FIRST(4), SU_GOOD(5), SU_BAD(6), NO_DATA(7)

d_met : decoding metric (max. 32767 = 0x7fff)
d_nerr : number of estimated errors (max. 196 = 0x00c4)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	blud	0	0	0	0	0	0	prefa	0	0	rx2	rx1	rx0	typ2	typ1	typ0
1	d_met															
2	d_nerr															
3	c1a (0)															c1a (15)
4	(16)															(31)
5	(32)															(47)
6	(48)												c1a (60)	c1b (0)	(1)	(2)
7	(3)															(18)
8	(19)															(34)
9	(35)															(50)
10	(51)	(52)	(53)	(54)	(55)	(56)	(57)	c1b (58)	c2 (0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
11	(8)															(23)
12	(24)	(25)	(26)	(27)	0	0	0	0	0	0	0	0	0	0	0	0
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

"-" : undefined.

9.1.14 Format for TCH/AHS6.7

Use : a_du_0 or a_du_1 according to subchannel (Uplink), a_dd_0 or a_dd_1 according to subchannel(Downlink)

Uplink :

d-bits :

b_blud : flag for block transmission.
typ(i) : b_amr_channel_type = AFS_6.7 (011)
rx(i) : unused

d_met : not used in encoder.
d_nerr : not used in encoder.

Downlink :

d-bits :

b_blud : block receive flag, set to 1 by AMR scheduler as default.
prefa : FACCH has been predetected(1)
typ(i) : b_amr_channel_type = AFS_6.7 (011)
rx(i) : b_amr_rx_type depending on the channel conditions : SPEECH_GOOD(0), SPEECH_DEGRADED(1), ONSET(2), SPEECH_BAD(3), SID_FIRST(4), SU_GOOD(5), SU_BAD(6), NO_DATA(7)

d_met : decoding metric (max. 32767 = 0x7fff)
d_nerr : number of estimated errors (max. 200 = 0x00c8)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	blud	0	0	0	0	0	0	prefa	0	0	rx2	rx1	rx0	typ2	typ1	typ0
1	d_met															
2	d_nerr															
3	c1a (0)															c1a (15)
4	(16)															(31)
5	(32)															(47)
6	(48)	(49)	(50)	(51)	(52)	(53)	c1a (54)	c1b (0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
7	(9)															(24)
8	(25)															(40)
9	(41)	(42)	(43)	(44)	(45)	(46)	(47)	(48)	(49)	(50)	(51)	(52)	(53)	c1b (54)	c2 (0)	(1)
10	(2)															
11	(18)	(19)	(20)	(21)	(22)	(23)	0	0	0	0	0	0	0	0	0	0
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

"-" : undefined.

9.1.15 Format for TCH/AHS5.9

Use : a_du_0 or a_du_1 according to subchannel (Uplink), a_dd_0 or a_dd_1 according to subchannel(Downlink)

Uplink :

d-bits :

b_blud : flag for block transmission.
typ(i) : b_amr_channel_type = AHS_5.9 (010)
rx(i) : unused

d_met : not used in encoder.
d_nerr : not used in encoder.

Downlink :

d-bits :

b_blud : block receive flag, set to 1 by AMR scheduler as default.
prefa : FACCH has been predetected(1)
typ(i) : b_amr_channel_type = AFS_5.9 (010)
rx(i) : b_amr_rx_type depending on the channel conditions : SPEECH_GOOD(0), SPEECH_DEGRADED(1), ONSET(2), SPEECH_BAD(3), SID_FIRST(4), SU_GOOD(5), SU_BAD(6), NO_DATA(7)

d_met : decoding metric (max. 32767 = 0x7fff)
d_nerr : number of estimated errors (max. 208 = 0x00d0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	blud	0	0	0	0	0	0	prefa	0	0	rx2	rx1	rx0	typ2	typ1	typ0
1	d_met															
2	d_nerr															
3	c1a (0)															c1a (15)
4	(16)															(31)
5	(32)															(47)
6	(48)						c1a (54)	c1b (0)								(8)
7	(9)															
8	(25)															(40)
9	(41)					c1b (46)	c2 (0)									(9)
10	(10)	(11)	(12)	(13)	(14)	(15)	0	0	0	0	0	0	0	0	0	0
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

"-" : undefined.

9.1.16 Format for TCH/AHS5.15

Use : a_du_0 or a_du_1 according to subchannel (Uplink), a_dd_0 or a_dd_1 according to subchannel(Downlink)

Uplink :

d-bits :

b_blud : flag for block transmission.
typ(i) : b_amr_channel_type = AFS_5.15 (001)
rx(i) : unused

d_met : not used in encoder.
d_nerr : not used in encoder.

Downlink :

d-bits :

b_blud : block receive flag, set to 1 by AMR scheduler as default.
prefa : FACCH has been predetected(1)
typ(i) : b_amr_channel_type = AFS_5.15 (001)
rx(i) : b_amr_rx_type depending on the channel conditions : SPEECH_GOOD(0), SPEECH_DEGRADED(1), ONSET(2), SPEECH_BAD(3), SID_FIRST(4), SU_GOOD(5), SU_BAD(6), NO_DATA(7)

d_met : decoding metric (max. 32767 = 0x7fff)
d_nerr : number of estimated errors (max. 212 = 0x00d4)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	blud	0	0	0	0	0	0	prefa	0	0	rx2	rx1	rx0	typ2	typ1	typ0
1	d_met															
2	d_nerr															
3	c1a (0)															c1a (15)
4	(16)															(31)
5	(32)															(47)
6	c1a (48)	c1b (0)														(14)
7	(15)															
8	(31)										c1b (41)	c2 (0)				(4)
9	(5)	(6)	(7)	(8)	(9)	(10)	(11)	0	0	0	0	0	0	0	0	0
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

"-" : undefined.

9.1.17 Format for TCH/AHS4.75

Use : a_du_0 or a_du_1 according to subchannel (Uplink), a_dd_0 or a_dd_1 according to subchannel(Downlink)

Uplink :

d-bits :

b_blud : flag for block transmission.
typ(i) : b_amr_channel_type = AHS_4.75 (000)
rx(i) : unused

d_met : not used in encoder.
d_nerr : not used in encoder.

Downlink :

d-bits :

b_blud : block receive flag, set to 1 by AMR scheduler as default.
prefa : FACCH has been predetected(1)
typ(i) : b_amr_channel_type = AHS_4.75 (000)
rx(i) : b_amr_rx_type depending on the channel conditions : SPEECH_GOOD(0), SPEECH_DEGRADED(1), ONSET(2), SPEECH_BAD(3), SID_FIRST(4), SU_GOOD(5), SU_BAD(6), NO_DATA(7)

d_met : decoding metric (max. 32767 = 0x7fff)
d_nerr : number of estimated errors (max. 212 = 0x00d4)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	blud	0	0	0	0	0	0	prefa	0	0	rx2	rx1	rx0	typ2	typ1	typ0
1	d_met															
2	d_nerr															
3	c1a (0)															c1a (15)
4	(16)															(31)
5	(32)						c1a (38)	c1b (0)								(8)
6	(9)															(23)
7	(25)															(40)
8	(41)	(42)	c1b (43)	c2 (0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	c2 (11)	0
9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

"-" : undefined.

9.1.18 Format for SID_UPDATE

Use : a_du_0 (Uplink), a_dd_0 (Downlink)

Uplink :

d-bits :

b_blud : flag for block transmission.
typ(i) : b_amr_channel_type (current channel type)
rx(i) : unused

d_met : not used in encoder.
d_nerr : not used in encoder.

Downlink :

d-bits :

b_blud : block receive flag, set to 1 by AMR scheduler as default.
typ(i) : b_amr_channel_type (current channel type)
rx(i) : b_amr_rx_type depending on the channel conditions (SID_UPDATE, SID BAD)

d_met : decoding metric (max. 32767 = 0x7fff)
d_nerr : number of estimated errors (max. 35 = 0x0023)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Blud	0	0	0	0	0	0	0	0	0	rx2	rx1	rx0	typ2	typ1	typ0
1	d_met															
2	d_nerr															
3	(0)															(15)
4	(16)															(31)
5	(32)	(33)	(34)	0	0	0	0	0	0	0	0	0	0	0	0	0
6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

"-" : undefined.

9.1.19 Format for RATSCCH

Use : a_ratscch_ul (Uplink), a_ratscch_dl (Downlink)

Uplink :

d-bits :

b_blud : flag for block transmission.
typ(i) : b_amr_channel_type (current channel type)
rx(i) : unused

d_met : not used in encoder.
d_nerr : not used in encoder.

Downlink :

d-bits :

b_blud : block receive flag, set to 1 by AMR scheduler as default.
typ(i) : b_amr_channel_type (current channel type)
rx(i) : b_amr_rx_type = depending on the channel conditions RATSCCH_GOOD (101) or RATSCCH_BAD (110)

d_met : decoding metric (max. 32767 = 0x7fff)
d_nerr : number of estimated errors (max. 35 = 0x0023)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	blud	0	0	0	0	0	0	0	0	0	rx2	rx1	rx0	typ2	typ1	typ0
1	d_met															
2	d_nerr															
3	(0)															(15)
4	(16)															(31)
5	(32)	(33)	(34)	0	0	0	0	0	0	0	0	0	0	0	0	0

"-" : undefined.

9.1.20 Format for TCH/F144

Use : a_du_0, a_du_1 (Uplink), a_dd_0, a_dd_1 (Downlink)

Uplink :

d-bits :

b_blud : flag for block transmission.

(*) Channel encoder copies this flag to the t-flags in the interleaver.

b_af : not used in encoder.

d_macc : not used in encoder.

d_nerr : not used in encoder.

Downlink :

d-bits :

b_blud : block receive flag, set to 1 by decoder as default.

b_af : set to 1 by decoder as default.

d_macc : accumulated metric (max. 25536 = 0x63c0)

d_nerr : number of estimated errors (max. 456 = 0x01c8)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	blud	af	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	d_macc															
2	d_nerr															
3	d(15)															d(0)
4	(31)															(16)
5	(47)															(32)
6	(63)															(48)
7	(79)															(64)
8	(95)															(80)
9	(111)															(96)
10	(127)															(112)
11	(143)															(128)
12	(159)															(144)
13	(175)															(160)
14	(191)															(176)
15	(207)															(192)
16	(223)															(208)
17	(239)															(224)
18	(255)															(240)
19	(271)															(256)
20	(287)															(272)
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	(289)	(288)

"-" : undefined.

9.1.21 Format for TCH/F96

Use : a_du_0, a_du_1 (Uplink), a_dd_0, a_dd_1 (Downlink)

Uplink :

d-bits :

b_blud : flag for block transmission.

(*) Channel encoder copies this flag to the t-flags in the interleaver.

b_af : not used in encoder.

d_macc : not used in encoder.

d_nerr : not used in encoder.

Downlink :

d-bits :

b_blud : block receive flag, set to 1 by decoder as default.

b_af : set to 1 by decoder as default.

d_macc : accumulated metric (max. 25536 = 0x63c0)

d_nerr : number of estimated errors (max. 456 = 0x01c8)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	blud	af	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	d_macc															
2	d_nerr															
3	d(15)															d(0)
4	(31)															(16)
5	(47)															(32)
6	(63)															(48)
7	(79)															(64)
8	(95)															(80)
9	(111)															(96)
10	(127)															(112)
11	(143)															(128)
12	(159)															(144)
13	(175)															(160)
14	(191)															(176)
15	(207)															(192)
16	(223)															(208)
17	(239)															(224)
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

"-" : undefined.

9.1.22 Format for TCH/F48

Use : a_du_0 (Uplink), a_dd_0 (Downlink)

Uplink :

d-bits :

b_blud : flag for block transmission.

(*)Channel encoder copies this flag to the t-flags in the interleaver.

b_af : not used in encoder.

d_macc : not used in encoder.

d_nerr : not used in encoder.

Downlink :

d-bits :

b_blud : block receive flag, set to 1 by decoder as default.

b_af : set to 1 by decoder as default.

d_macc : accumulated metric (max. 25536 = 0x63c0)

d_nerr : number of estimated errors (max. 456 = 0x01c8)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	blud	af	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	d_macc															
2	d_nerr															
3	d(15)															d(0)
4	(31)															(16)
5	(47)															(32)
6	(63)															(48)
7	(79)															(64)
8	(95)															(80)
9	(111)															(96)
10	0	0	0	0	0	0	0	0	(119)							(112)
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

"-" : undefined.

9.1.23 Format for TCH/F24

Use : a_du_0 (Uplink), a_dd_0 (Downlink)

Uplink :

d-bits :

b_blud : flag for block transmission.

(*)Channel encoder copies this flag to the t-flags in the interleaver.

b_af : not used in encoder.

d_macc : not used in encoder.

d_nerr : not used in encoder.

Downlink :

d-bits :

b_blud : block receive flag, set to 1 by decoder as default.

b_af : set to 1 by decoder as default.

d_macc : accumulated metric (max. 25536 = 0x63c0)

d_nerr : number of estimated errors (max. 456 = 0x01c8)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	blud	af	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	d_macc															
2	d_nerr															
3	d(15)															d(0)
4	(31)															(16)
5	(47)															(32)
6	(63)															(48)
7	0	0	0	0	0	0	0	0	(71)							(64)
8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

"-" : undefined.

9.1.24 Format for FACCHF/H and CCH

Use : a_fu, a_cu (Uplink), a_fd, a_cd (Downlink)

a_fu contains an info frame for FACCH/F and FACCH/H (uplink)
a_cu contains an info frame for CCH (uplink)
a_fd contain an info frame for FACCH/F and FACCH/H (downlink)
a_cd contain an info frame for CCH (downlink)

Uplink :

d_bits :

b_blud : “1” info frame available, “0” no info frame.
b_fire0, b_fire1, b_facch_det : not used in encoder.

d_macc : not used in encoder.
d_nerr : not used in encoder.

Downlink :

d_bits :

b_blud : set to 1 by decoder when a block is decoded.
b_fire1, b_fire0 : fire flags
(fire1,fire0) : (0, 0) no error detected
(0, 1) error detected and corrected
(1, 0) erroneous frame detected
b_facch_det : “0” no FACCH detected, “1” FACCH detected.

d_macc : accumulated metric
FACCHF :max. 21169 = 0x52b1
FACCHH : max. 23620 = 0x5c44
SACCH : max. 25536 = 0x63c0 (no normalization)

d_nerr : number of estimated errors (max. 456 = 0x01c8)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	blud	0	0	0	0	0	0	0	fcdet	fire1	fire0	0	0	0	0	0
1	d_macc															
2	d_nerr															
3	d(15)															d(0)
4	(31)															(16)
5	(47)															(32)
6	(63)															(48)
7	(79)															(64)
8	(95)															(80)
9	(111)															(96)
10	(127)															(112)
11	(143)															(128)
12	(159)															(144)
13	(175)															(160)
14	0	0	0	0	0	0	0	0	(183)							(176)

9.1.25 Format for RACH

Use : d_rach (Uplink)

- Info byte d(0) - d(7)
- Base station identity code BSIC b(0) - b(5) (b(5) is MSB)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
d(7)	d(6)	d(5)	d(4)	d(3)	d(2)	d(1)	d(0)	b(5)	b(4)	b(3)	b(2)	b(1)	b(0)	0	0

9.1.26 Format for SCH

Use : a_sch (Downlink)

d_bits :

b_blud : set to 1 by decoder as default.

b_sch_crc : status of the crc decoder, “0” no error detected, “1” error detected.

d_macc : accumulated metric (max. 4368 = 0x1110)

d_nerr : set to 0 by decoder. (no reconvolution)

a_frame : frame information

d(0), d(1) : T1 high
d(2)-d(7) : BSIC
d(8)-d(15) : T1 middle
d(16),d(17) : T3’ high
d(18)-d(22) : T2
d(23) : T1 low
d(24) : T3’ low

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	blud	0	0	0	0	0	0	sch crc	0	0	0	0	0	0	0	0
1	d_macc															
2	d_nerr															
3	d(15)															d(0)
4	0	0	0	0	0	0	0	d(24)								d(16)

9.1.27 Access of the NDB Area fields

It concerns the header of each buffer :

for a_cd, a_fd, a_dd_0, a_dd_1 : MCU will reset the header after reading
for a_cu, a_fu, a_du_0, a_du_1 : DSP will reset the header after reading

Here-below tables are giving the **d_fn_report** frame number at which data buffers are **accessed** by DSP for TCH modes.

Mode	Buffer	d_fn_report
TCH/ FS / EFS /AMR FS	a_dd_0	11,3,7 (modulo 13)
TCH/F2.4	a_dd_0	11,3,7 (modulo 13)
TCH/HS0/AMR HS0	a_dd_0	2, 6, 10 (modulo 13)
TCH/HS1/AMR HS1	a_dd_1	3, 7, 11 (modulo 13)
TCH/F4.8 – TCH/F9.6 – TCH/F14.4	a_dd_0	22,1,5,9,14,18 (26 Modulo)
FACCH/F(1)	a_fd	11,3,7 (13 modulo)
SACCH/TF	a_cd	90
FACCH/HS0 (2)	a_fd	2, 6, 10 (modulo 13)
FACCH/HS1 (2)	a_fd	3, 7, 11 (modulo 13)
SACCH/TH	a_cd	90

DSP Access for Downlink Buffers in TCH modes

Note :

1- When an FACCH is detected , the a_fd header is copied in a_dd_0 buffer. This is done for any TCH mode. Reason for this is to control the speech regulation algorithm with d_macc value when FACCH is found.
For full rate datas , same operation is done so that blud bit of a_dd_0 buffer can be set to 1 at boundaries 11,3,7 whereas it is expected to toggle only at 22,1,5,9,14,18 boundaries.

2- Above comment is also applicable to half rate.

Mode	Buffer	d_fn_report
TCH/ FS	a_du_0	11,3,7 (13 Modulo)
TCH/ EFS	a_du_0	10,2,6 (13 Modulo)
TCH/F2.4	a_du_0	11,3,7 (13 Modulo)
TCH/HS0	a_du_0	2, 6, 10 (modulo 13)
TCH/HS1	a_du_1	3, 7, 11 (modulo 13)
TCH/F4.8 - TCH/F9.6 – TCH/F14.4	a_du_0	24,3,7,11,16,20 (26 Modulo)
FACCH/F	a_fu	11,3,7 (13 Modulo)
SACCH/TF	a_cu	12
SACCH/TH	a_cu	12

DSP Access for Uplink Buffers in TCH modes

For CCH,PCH,BCCH, .. modes , a_cd and a_cu buffers are used. a_cd buffer is written by DSP when burst number d_burst_d is 3 and a_cu buffer is read by DSP when burst number d_burst_u is 0.

9.1.28 Range for d_nerr, d_macc in Header

d_nerr corresponds to the estimated number of errors before channel decoder (demodulator output). This value is computed by making convolutionnal encoding of decisions from channel decoder and then counting of "errors" between these bits and decisions (converted to hard decisions) from demodulator. For a given mode , the higher d_nerr is , the higher the estimated error rate is. This information is used by L1 upper level in order to assert RXQUAL (GSM5.08).

d_nerr is provided for all types of traffics except SCH for which it is not necessary and would be a CPU penalty. Here-below table is giving the maximum d_nerr value as function of the type of information. Basically the possible range of d_nerr is related to the number of convolutionnaly encoded bits for the given mode. Basic process in L1 is to sum d_nerr values and then normalize with this maximum value prior formatting to GSM spec format (see GSM5.08). Note: maximum value for d_nerr is never reached . As rough indication, 35% of this value is reached when very bad conditions are encountered.

d_macc is provided for all types of traffics . This value corresponds to the accumulated metric found for best decided path during channel decoding. Th higher the metric is , "best" is the path. d_macc is used by DSP in its bad frame indication process as well as for speech regulation. For speech regulation, this value is used to control volume at speaker output. Here-below table is giving the maximum d_macc value as function of the type of information. Basically the possible range of d_macc is related to the number of convolutionnaly encoded bits for the given mode.

Mode	d_macc_Max Value	d_nerr Max Value
TCH/FS	21168	378
TCH/HS	11816	211
TCH/F2.4	25536	456
TCH/F9.6 or TCH/H4.8 or TCH/F14.4 (1)	25536	456
TCH/F4.8 or TCH/H2.4	25536	456
FACCH/F(2)	21169	456
FACCH/H(3)	23632	456
BCCH/CCCH/SDCCH/SACCH	25536	456
SCH	#4674	NA

(1) These channels are using puncturing. Of course, punctured values are not taken into account in the computation of d_nerr and d_macc.
(2) This maximum value is obtained after normalizing 25536 distance by 378/456 factor. This normalization is done in order that the distance dynamic for FACCH/F is same as for TCH/FS. This is needed to input FACCH/F distance into speech regulation filter.
(3) This maximum value is obtained after normalizing 25536 distance by 211*2/456 factor. This normalization is done in order that the distance dynamic for FACCH/H is two times as for TCH/FS.
This is needed to input FACCH/H distance into speech regulation filter (factor two is needed because one FACCH/H block is stealing two TCH/HS speech frames).

9.1.29 Details for Voice Memo MMS-AMR operation

9.1.29.1 Control bitfield

Refer to d_tch_mode, d_tonesk_init and d_toneskb_status description

9.1.29.2 a_du_1 data buffer specific to MMS-AMR operation

This buffer is used to transfer encoded speech frames between the MCU and the DSP.

The following bit-fields are defined into the ***a_du_1*** buffer header.

- **b_amms_rx_type :** three info bits {rxtype[2], rxtype[1], rxtype[0]}
  This code corresponds to the fields RX_TYPE described in the GSM 06.93
- **b_amms_tx_type :** three info bits {txtype[2], txtype[1], txtype[0]}
  This code corresponds to the fields TX_TYPE described in the GSM 06.93
- **b_amms_channel_type :** three info bits {chan_type[2], chan_type[1], chan_type[0]}

These bit-fields are coded as the following:

Binary code	b_amms_channel_type	b_amms_rx_type	b_amms_tx_type
000	4.75	SPEECH_GOOD	SPEECH_GOOD
001	5.15	SPEECH_DEGRADED (1)	*Unused*
010	5.9	ONSET (1)	*Unused*
011	6.7	SPEECH_BAD (1)	*Unused*
100	7.4	SID_FIRST	SID_FIRST
101	7.95	SID_UPDATE	SID_UPDATE
110	10.2	SID_BAD (1)	*Unused*
111	12.2	NO_DATA	NO_DATA

(1) SPEECH DEGRADED, ONSET, SPEECH_BAD and SID_BAD types are used only in play mode. They are managed by the MCU layer 1 according to ETSI 0693 and ETSI 05.03.

In record mode, the header of the *a_du_1* buffer must be cleared by the MCU a frame has been read.

In record mode, frames must be processed by MCU to allow a correct later playback. This processing is mandatory whenever a transition DTX->SPEECH occurs. It consist in inserting an ONSET frame (which will add a 20ms delay) between the “DTX frame” and the speech frame, this indicates the speech decoder that DTX mode is over.
For instance, if the recorded sequence is :

```
Frame   1    2    3    4    5    6    7    8    9  ...
| SP | SP | SF | ND | ND | SU | ND | ND | SP | SP |...
```

the pre-processing should give the following sequence to play:

```
| SP | SP | SF | ND | ND | SU | ND | ND | ON| SP | SP |…
```

Actually, whenever a SPEECH/SP DEGRADED/SP BAD frame has to be played, it must be preceded by an **ONSET** if previous frame is different from SP/SP DEGRADED/SP BAD. Consequently if the first frame to be played is a speech frame, an ONSET must be inserted before.

Example of data bits allocations for the AFS12.2 vocoder, see the S825.DOC for the other data rates.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	blud	0	0	0	0	block_ready	0	0	0	0	rx/tx_type2	rx/tx_type 1	rx/tx_type 0	chan_type 2	chan_type 1	chan_type 0
1	0															
2	0															
3	c1a (0)															c1a (15)
4	(16)															(31)
5	(32)															(47)
6	(48)															(63)
7	(64)															(79)
8	c1a	c1b														c1b

	(80)	(0)														(14)
9	c1b (15)															c1b (30)
10	(31)															(46)
11	(47)															(62)
12	(63)															(78)
13	(79)															(94)
14	(95)															(110)
15	(111)															(126)
16	(127)															(142)
17	(143)															(158)
18	(159)	(160)	(161)	(162)	0	0	0	0	0	0	0	0	0	0	0	0
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

"-" : undefined.

(*) bit 10 is b_block_ready. In record mode, this bit is used by the DSP to indicate to the MCU that a encoded speech block is available for storing. In play mode, this is used by the MCU to indicate to the DSP that an encoded speech block is available for decoding.

9.1.29.3 Recording additional interface

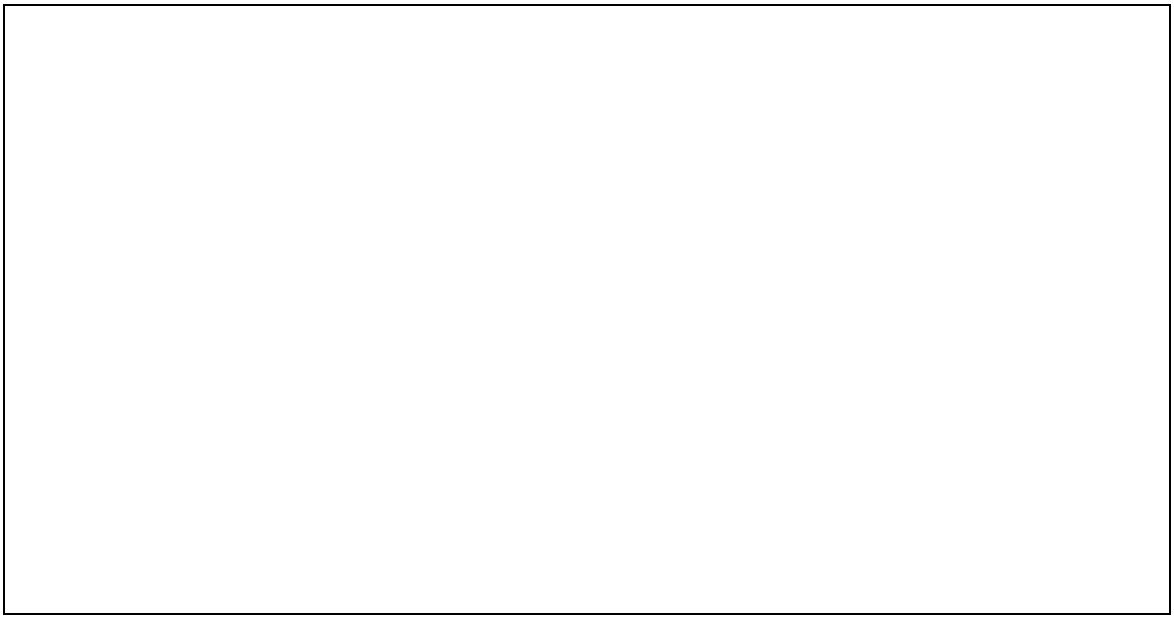
d_amms_ul_voc

Bits0-2: b_amms_channel_type: Selects the AMR codec to be used during the dictaphone recording.

Same format as described above.

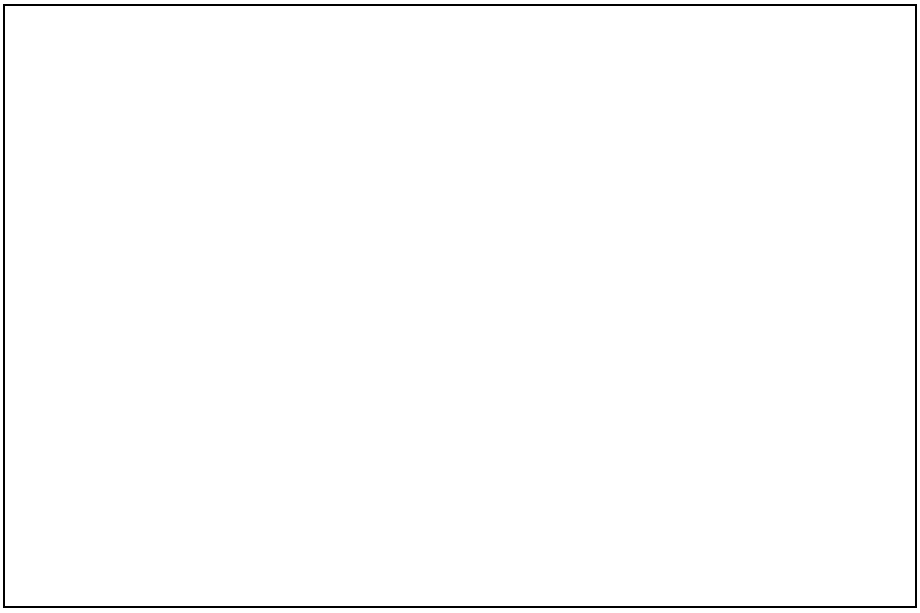
9.2 Power Measurements (all modes)

Functional Description : DSP performs **0,1,2 or 3** measurements inside communication period. If it is the only task to perform within the communication period, the downlink and uplink tasks are set to NO_TASK.
Communication Period : 1 "TDMA" (Ctrl-Wrk-Rd)
Number of periods : Measurements window



9.3 BCCH Tasks for neighbor cell (Idle mode)

Functional Description : DSP performs demodulation / decoding for a bcch channel (rectangular 4 interleaver). No power measurements should be requested in this case.
Communication Period : 5 "TDMAs" (Ctrl- 5 Wrk - Rd).
Number of periods : 1 period for decoding of a sysinfo block

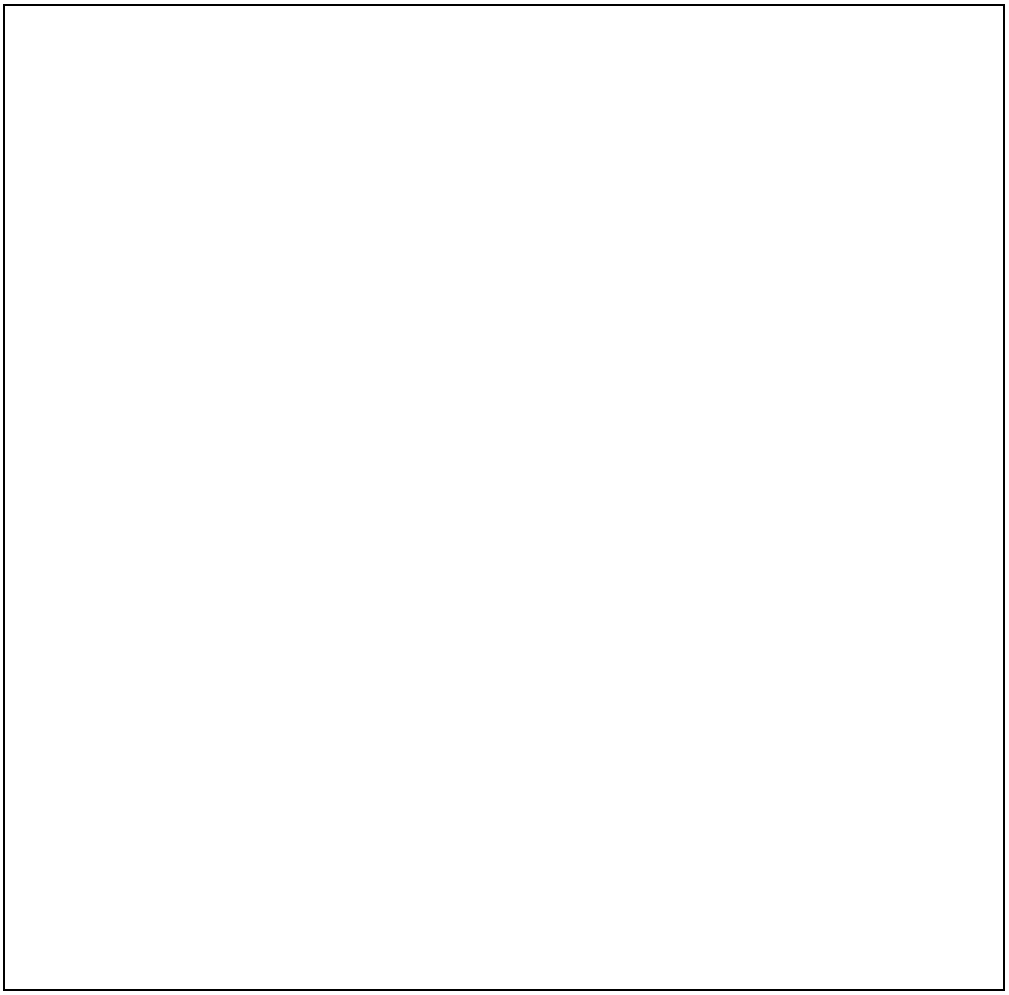


9.4 PCH, BCCH, AGCH Tasks for serving cell (Idle mode)

Functional Description : DSP performs demodulation / decoding for a bcch, pch, agch like channel (rectangular 4 interleaver). The “d_task_md” parameter allows to combine measurements within same period.

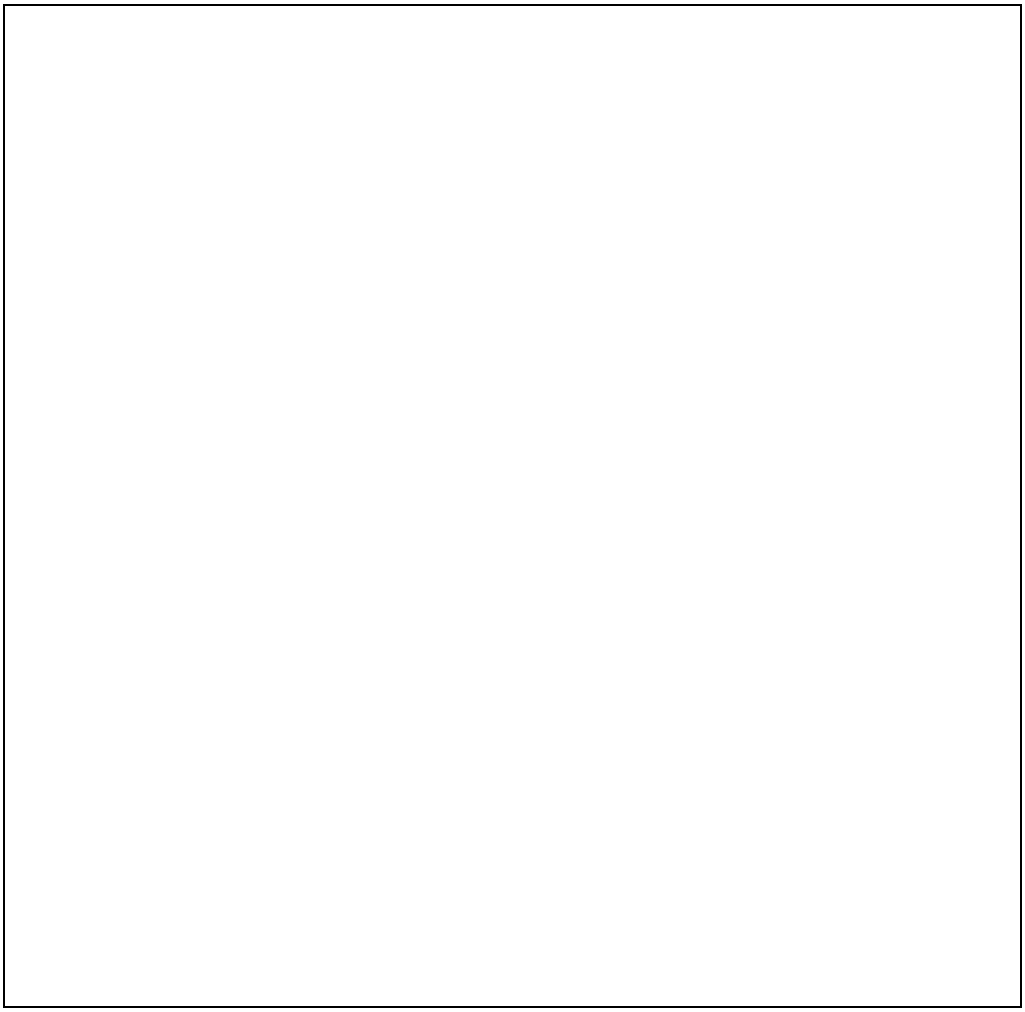
Communication Period : 1 "TDMA". In this case the MCU request is on a “burst” basis. (Ctrl - Wrk - Rd)

Number of periods : 4 periods for decoding of a pch/bcch/agch block



9.5 SDCCH, SACCH Tasks (SDCCH Mode)

Functional Description : DSP performs demodulation / decoding for a sdcch/sacch like channel (rectangular 4 interleaver). The “d_task_md” parameter allows to combine measurements within same period.
Communication Period : 1 "TDMA". In this case the MCU request is on a “burst” basis. (Ctrl - Wrk - Rd)
Number of periods : 4 periods for decoding of a sdcch/sacch block



9.6 TCH Tasks

Functional Description : DSP performs demodulation / decoding for a traffic channel. “d_task_md” parameter allows to combine measurements within same period, “d_fn_report” gives the FN within 104MF.
Communication Period : 1 "TDMA". (Ctrl- Wrk-Rd)

The code reserved for SACCH in TCH mode is TCHA_TASK = 14. The code reserved for normal burst in TCH mode is TCHT_TASK = 13. For TCH/HS the unused TCH frames are dummy TCH: TCHD_TASK = 28.

10 DETAILS ON THE INTERFACE – GPRS only

10.1 Layer-1 Scenario and results

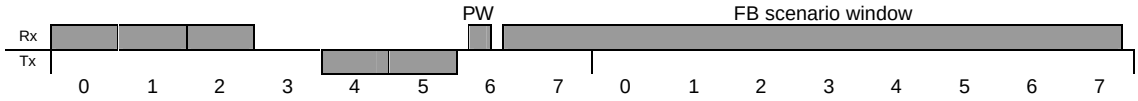
This paragraph presents how L1 scenarios are set into the DB area, where DSP scheduler take input buffers and where results are returned to the MCU. Basic concepts are figured by an example, then a detailed explanation is done for each scenario. Moreover, some special cases are described.

10.1.1 General case

10.1.1.1 Scenario

GPRS basis scenarios are presented by the Layer-1 using the *d_task_u_gprs, d_task_d_gprs and d_task_pm_gprs* words. Basis scenarios include normal burst downlink, normal burst uplink, PTCCH uplink and downlink, access burst and power monitoring. Frequency burst monitoring, synchronization burst monitoring and Interference measurement scenarios are presented using the common GSM interface: *d_task_md* word in DB area.

The following example presents a multislot class 12 mobile with a two TDMAs scenario work period:



Bit	d_task_u_gprs	d_task_d_gprs	d_task_pm_gprs	Timeslot	Scenario	Observation
7	0	1	0	TS 0	Downlink	Three contiguous NB.
6	0	1	0	TS 1	Downlink	
5	0	1	0	TS 2	Downlink	
4	0	0	0	TS 3	No Task	DMA can be reprogrammed
3	1	0	0	TS 4	Uplink	Two contiguous NB.
2	1	0	0	TS 5	Uplink	
1	0	0	1	TS 6	PW Monitoring	PW can be in the middle of two TS.
0	0	0	0	TS 7	No Task	

d_task_md	TCH_FB_TASK	Around 9 physical timeslots duration.

The "timeslot" concept used in the GPRS C Scheduler code permits to know if scenarios are contiguous or not. The DMA programming is allowed between two non-consecutive downlink windows.

Normal burst uplink scenario (d_task_u_gprs)

The normal burst uplink scenario is set by the MCU to indicate exactly where one NB will be sent. Up to eight NB uplink tasks can be requested in the TDMA. The element of *a_ul_buffer_gprs* array corresponding to the scenario (same "timeslot" area) designates the uplink buffer, it allows to chose between Poll-response buffer and uplink data buffer. Moreover, the RLC/MAC buffer CS_TYPE field is used to know which CS must be used to encode the buffer.

Access burst (PRACH) scenario (d_task_u_gprs)

The access burst scenario is set by the MCU to indicate exactly where the access burst will be sent. The element of *a_ul_buffer_gprs* array corresponding to the scenario (same "timeslot" area) designates the uplink buffer: Uplink data buffer (*a_du_gprs*) or Poll-response buffer (*a_pu_gprs*). Moreover, the RLC/MAC buffer CS_TYPE field is used to select between 8bit and 11bit PRACH to encode the buffer. In case of access burst send during the access phase, the MCU must conjointly set the bit *b_access_prach* to 1 in the *d_task_u_gprs*word.

Normal burst downlink scenario (d_task_d_gprs)

The normal burst downlink scenario is set by the MCU to indicate exactly where one NB is intended to be received. Up to eight NB downlink tasks can be requested in the TDMA. The results are returned to the MCU using *d_burst_toa_gprs*, *d_burst_pm_gprs*, *d_burst_angle_gprs*, *d_burst_snr_gprs* of the corresponding timeslot area in DB and decoded burst are stored sequentially into *a_dd_gprs* buffers. The DSP return the CS type into the RLC/MAC buffer CS_TYPE field.

PW monitoring scenario (d_task_pm_gprs)

The power monitoring scenario is set by the MCU to indicate the timeslot where the PW measurement start, but the physical window can be executed later. For example, it can be executed in the middle of two physical timeslots accordingly to the TPU programmation. Up to eight PW monitoring tasks can be requested in the TDMA. The result of power monitoring module is returned to the MCU using the *d_burst_pm_gprs* in the corresponding timeslot area in DB.

Uplink PTCCH burst scenario (d_task_u_gprs)

Uplink PTCCH burst is send thanks to PRACH during the IDLE frame. The bit *b_ptcch_u* located into *d_task_u_gprs* is set to 1 concurrently to the Timeslot location information. *a_ul_buffer_gprs* is not used to select uplink buffer. The *a_ptcchu_gprs* buffer is used as input.

Downlink PTCCH burst scenario (d_task_d_gprs)

The downlink PTCCH is received on 4 IDLE frame TDMAs. The bit *b_ptcch_d* located into *d_task_d_gprs* is set to 1 concurrently to the Timeslot location information. The results are returned to the MCU using *d_burst_toa_gprs*, *d_burst_pm_gprs*, *d_burst_angle_gprs*, *d_burst_snr_gprs* of the corresponding timeslot area in DB and decoded burst are stored into *a_dd_md_gprs* buffer. CS type is forced to CS1 to decode the block.

Synchronization burst scenario (d_task_md)

Synchronization burst scenario is set by the MCU using the same interface than GSM (*d_task_md* word into DB area).
In packet transfer mode, the synchronization burst scenario has a 9 physical timeslots window length. Results (toa, pm, angle and snr) are returned to the MCU using the standard GSM FB/SB area (*d_fb_toa, d_fb_pm, d_fb_angle* and *d_fb_snr*) and *a_sch* area both in NDB.

For a SB acquisition in packet IDLE mode, the synchronization burst scenario can be set two times on two consecutive TDMAs. Then results are returned using the DB area instead of NDB: *d_toa*, *d_pm*, *d_angle*, *d_snr*, *SCH1*, *SCH2*, *SCH3*, *SCH4* and *SCH5*.

Frequency burst scenario (d_task_md)

Frequency burst scenario is set by the MCU using the same interface than GSM (*d_task_md word* into DB area).
In packet transfer mode, the frequency burst scenario has around 9 physical timeslots window length. Results (toa, pm, angle and snr) are returned to the MCU using the standard GSM area: *d_fb_toa*, *d_fb_pm*, *d_fb_angle*, *d_fb_snr* and *d_fb_det* words in NDB.

In packet IDLE mode, the frequency burst scenario has around 12 TDMAs physical window length. Results are returned to the MCU using the standard GSM area.

Neighbor BCCH monitoring scenario (d_task_md)

This scenario is set by the MCU using the d_task_md word. This scenario is performed on 5 work periods (C|W|W|W|W|R). The DSP use an internal counter to know which burst is received, and when the channel decoder must be performed.
The demodulator results are stored into d_toa, d_pm, d_angle and d_snr into the GSM DB area, then the Layer 1 will only access to the latest values (last burst). The buffer a_dd_gprs[0] is used to return the decoded block.

10.1.1.2 Buffers management

In order to optimize the memory resource, the number of uplink buffers and downlink buffers are defined for a specified multislot class number:

- For a multislot class 1: 1**a_du_gprs* buffer and 1**a_dd_gprs* buffer.
- For a multislot class 12: 4**a_du_gprs* buffers and 4**a_dd_gprs* buffers.

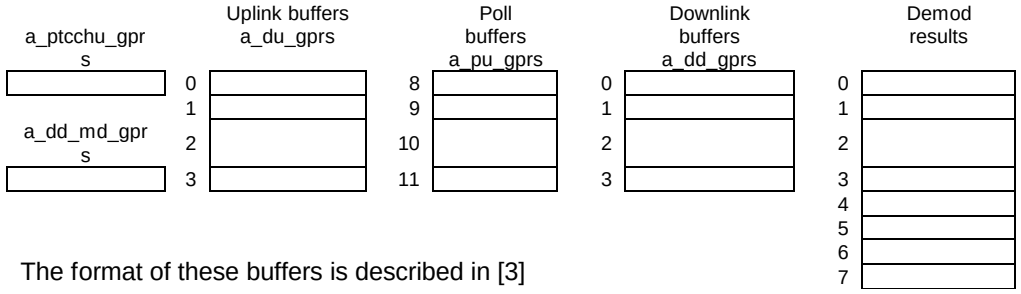
The same number as uplink buffers are reserved for Poll response uplink burst.

The demodulation measurements are stored into 8 areas to allow up to 8 receive bursts. Each area is constituted by 4 words: *d_burst_toa_gprs*, *d_burst_pm_gprs*, *d_burst_angle_gprs* and *d_burst_snr_gprs*. These areas are also used to store Power monitoring result (in *d_burst_pm_gprs*).

Uplink buffers, including Poll buffers, are indexed by an element of the *a_ul_buffer_gprs* array from NDB area. There are 8 words into *a_ul_buffer_gprs*, one per timeslot. Each element of *a_ul_buffer_gprs* can take a value into the range [0 to 7] for Uplink buffers and [8 to 15] for Polling buffers. For a multislot class 12 mobile, values into the range 4 to 7 and 12 to 15 are unused.

Downlink Buffers are stored sequentially. For a multislot class 12 mobile, values into the range 4 to 7 are unused. The demodulation results area is indexed with the same index than *d_task_d_gprs*.

The following figure presents interface for a multislot class 12 where 4*Rx and 4*Tx can be allocated:



The format of these buffers is described in [3]

10.1.1.3 Buffer format

a_du_gprs	Uplink data buffer
d_block_status	Bit [3-0]: b_cs_type 0000 None 0001 Not applicable 0010 CS1 - Data or control block 0011 Not applicable 0100 CS2 0101 CS3 0110 CS4 0111 PRACH 8 bits 1000 PRACH 11 bits Bit [5]: b_byte_shift If set to one, the first byte (LSB) of first word is not used.
a_data[28]	CS1: 12w (or 13w with byte shift). CS2: 17w (or 18w with byte shift). CS3: 20w (or 21w with byte shift). CS4: 27w (or 28w with byte shift). PRACH 8bits: 1w (or 2w with byte shift). PRACH 11 bits: 2w (or 3w with byte shift).

a_pu_gprs	Polling buffer

d_block_status	Bit [3-0]: b_cs_type 0000 None 0001 Not applicable 0010 Not applicable 0011 CS1 - Poll response 0100 Not applicable 0101 Not applicable 0110 Not applicable 0111 PRACH 8 bits 1000 PRACH 11 bits Bit [5]: b_byte_shift If set to one, the first byte (LSB) of first word is not used.
d_timeslot_hole	DSP does not used this word
a_data[13]	CS1: 12w (or 13w with byte shift). PRACH 8bits: 1w (or 2w with byte shift). PRACH 11 bits: 2w (or 3w with byte shift).

a_dd_gprs	Downlink data buffer
d_block_status	Bit [3-0]: b_cs_type 0000 None 0001 Not applicable 0010 CS1 - Data or control block 0011 Not applicable 0100 CS2 0101 CS3 0110 CS4 0111 Not applicable 1000 Not applicable Bit [8]: b_crc_error If set to one, Bad CRC. Bit [10-9]: b_tfi_result 00 TFI filtering not performed 01 Not applicable 10 Not applicable 11 Not applicable
d_timeslot_hole	DSP does not fill this word
d_macc	Accumulated metric (reported by the channel decoder) max is 456*56=25536.
d_nerr	Number of estimated errors (max=456), reported by the channel decoder.
a_data[27]	CS1: 12w. CS2: 17w. CS3: 20w. CS4: 27w.

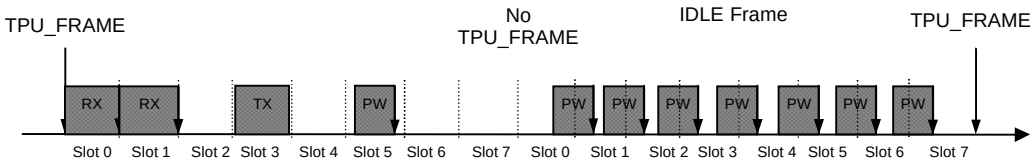
a_ptcchu_gprs	Uplink PTCCH buffer
d_block_status	Bit [3-0]: b_cs_type 0000 None 0001 Not applicable 0010 Not applicable 0011 Not applicable 0100 Not applicable 0101 Not applicable 0110 Not applicable 0111 PRACH 8 bits 1000 PRACH 11 bits Bit [5]: b_byte_shift If set to one, the first byte (LSB) of first word is not used.

a_data[3]	PRACH 8bits: 1w (or 2w with byte shift). PRACH 11 bits: 2w (or 3w with byte shift).

a_dd_md_gprs	Downlink PTCCH buffer
d_block_status	Bit [3-0]: b_cs_type 0000 None 0001 Not applicable 0010 CS1 - Data or control block 0011 Not applicable 0100 Not applicable 0101 Not applicable 0110 Not applicable 0111 Not applicable 1000 Not applicable Bit [8]: b_crc_error If set to one, Bad CRC. Bit [10-9]: b_tfi_result 00 TFI filtering not performed 01 Not applicable 10 Not applicable 11 Not applicable
d_timeslot_hole	DSP does not fill this word
d_macc	Accumulated metric (reported by the channel decoder) max is 456*56=25536.
d_nerr	Number of estimated errors (max=456), reported by the channel decoder.
a_data[12]	CS1: 12w.

10.1.2 Interference measurement

Interference measurements are performed during an IDLE frame and up to eight power monitoring tasks can be requested. Moreover, the MCU does not activate the communication interrupt for this scenario, it means that scenario must be presented during the previous TDMA. The following figure presents interference measurement with 7 power monitoring tasks:



The MCU use the *d_task_md* word in DB area to request such scenario, this word can be set to a value into the range [0 to 111]. Values for interference measurement are into the range [101 to 108]. The value indicates the number of power measurements that must be performed during the IDLE frame. The array *a_interference_meas_gprs* in the NDB is used to return results to the MCU, this array is filled from its first element and all power results are stored contiguously.

10.2 TPU driver constraints for GPRS

In case of 1Rx/1Tx config, the GPRS scheduler does not add any constraints on GSM TPU driver.

For multislot application, GPRS TPU driver must take in consideration the following constraints:

- Multislot Rx: The TPU receive window must have a 156 bits width for each normal burst.
- Multislot Tx: The INT9n DSP interrupt must be activated after the end of each ramp up/down. The TPU programmable interrupt is linked to the INT9n DSP interrupt.

10.3 Critical real time (USF vote)

- *d_usf_updated_gprs* word is used to report the USF state from the DSP to the MCU.

bit 15:14	bit 13:12	bit 11:10	bit 9:8	bit 7:6	bit 5:4	bit 3:2	bit 1:0
TS 0	TS 1	TS 2	TS 3	TS 4	TS 5	TS 6	TS 7

Two bits are used per timeslot and they can have the following values. These values are the USF state reported by the DSP to the MAC layer into the MCU:

00 – invalid

This means that the USF decoding has not been made (this is the default value)

01 – valid and good USF

This means that the USF decoding has been done and that the USF value found allows us to transmit on the next TDMA.

10 – valid and bad USF

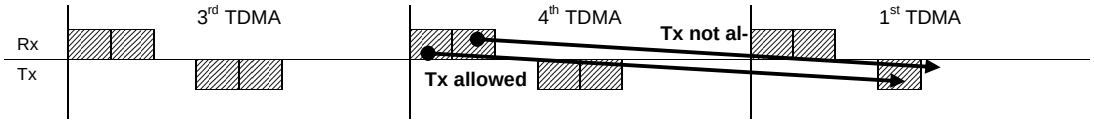
This means that the USF decoding has been done and that the USF quality is too bad or that the USF value does not allow us to transmit on the next TDMA.

Each timeslot state of *d_usf_updated_gprs* is set as soon as the DSP has finished the decoding of the current timeslot.
This word is filled only on the last radio block (TDMA 3) by the DSP and it is used on the first radio block (TDMA 0) by the MCU. This means that it can be reset to the default values only on the second or the third radio block (TDMA 1 or 2).

- *d_task_u_gprs* word is used by the MCU to report the USF result read by the MAC layer to the DSP. This word is used only on the first radio block.

Bit 15:8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
not used	TS 0	TS 1	TS 2	TS 3	TS 4	TS 5	TS 6	TS 7

One bit is used per timeslot and has the following values:
0 – MAC has not set the transmit (default value)
1 – MAC has set the transmit (buffer and TPU window)



- *d_usf_vote_enable* word is set by the MCU to enable/disable the DSP USF mechanism, especially to support USF granularity.

Bit 15:8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
not used	TS 0	TS 1	TS 2	TS 3	TS 4	TS 5	TS 6	TS 7

One bit is used per timeslot and has the following values:
0 – USF mechanism disabled.
1 – USF mechanism enabled.

11 DRP programming techniques

The DRP register programming is based on DB words that indicate when new values are needed to be sent to the DRP. The DB word depends on the mode:

- In GSM mode the d_ctrl_abb word is a bit field that contains AFC, APCDEL, APCDEL2 and APCRAM programming information. APCLEV is programmed each time a Tx is requested.
- The array a_ctrl_abb_gprs is used for AFC, APCRAM, APCDEL and APCDEL2 programming information. APCLEV is programmed each time a Tx is requested.

All these variables have the format corresponding to DRP/wrapper registers.

The next array resumes all API variables with their respective format and the way to indicate to the DSP when a new value needed to be considered.

DRP Register	API word		API Area	API Format	When DSP consider it (1)		Comment
APCLEV	GSM	d_power_ctrl	DB	[15-0] Value	On each Tx request		Formatted by ARM
	GPRS	a_ctrl_power_gprs[ts]					
APCRAM	GSM	a_drp_ramp[20]	NDB	[15-8] DW [7-0] UP	GSM	d_ctrl_abb.b_ramp=1	Formatted by ARM
	GPRS	a_drp_ramp2_gprs_0[20] a_drp_ramp2_gprs_1[20] a_drp_ramp2_gprs_2[20] a_drp_ramp2_gprs_3[20] a_drp_ramp2_gprs_4[20] a_drp_ramp2_gprs_5[20] a_drp_ramp2_gprs_6[20] a_drp_ramp2_gprs_7[20]			GPRS	a_ctrl_abb_gprs[ts].b_ramp=1	
APCDEL	d_apcdel		NDB	15-0	GSM	d_ctrl_abb.b_apcdel=1	Formatted by ARM
					GPRS	a_ctrl_abb_gprs[ts].b_apcdel=1 (2)	
APCDEL2	d_apcdel2		NDB	15-0	GSM	d_ctrl_abb.b_apcdel2=1	Formatted by ARM
					GPRS	a_ctrl_abb_gprs[ts].b_apcdel2=1 (2)	
DCXO_XTAL	d_afc		DB	[15-0] Value	d_ctrl_abb.b_afc=1		Formatted by ARM

Notes:

(1) [ts] designate the timeslot number into the range [0 to 7].

(2) Bits b_apcdel_bis and b_apcdel2_bis can also be used to reprogram APCDEL and APCDEL2 DRP wrapper registers through d_apcdel_bis and d_apcdel2_bis NDB words. These bits are located into a_ctrl_abb_gprs array and are also marked as reserved (must be set to 0) into d_ctrl_abb word.

Appendices

A. Acronyms

B. Tri-tones generator

k_tx (decimal) F0.8	Frequency (Hz)	Correction factor	k_tx (decimal) F0.8	Frequency (Hz)	Correction factor
40	1800,24	0,988	148	1215,14	0,816
41	1795,20	0,987	149	1209,03	0,813
42	1790,16	0,986	150	1202,91	0,810
43	1785,12	0,986	151	1196,76	0,808
44	1780,07	0,985	152	1190,59	0,805
45	1775,02	0,984	153	1184,40	0,802
46	1769,97	0,984	154	1178,18	0,799
47	1764,91	0,983	155	1171,94	0,796
48	1759,85	0,982	156	1165,68	0,793
49	1754,78	0,982	157	1159,40	0,790
50	1749,71	0,981	158	1153,09	0,787
51	1744,64	0,980	159	1146,75	0,784
52	1739,56	0,979	160	1140,40	0,781
53	1734,48	0,978	161	1134,01	0,777
54	1729,39	0,977	162	1127,60	0,774
55	1724,30	0,977	163	1121,16	0,771
56	1719,21	0,976	164	1114,70	0,768
57	1714,11	0,975	165	1108,21	0,765
58	1709,00	0,974	166	1101,69	0,761
59	1703,90	0,973	167	1095,14	0,758
60	1698,78	0,972	168	1088,57	0,755
61	1693,66	0,971	169	1081,96	0,751
62	1688,54	0,970	170	1075,32	0,748
63	1683,41	0,969	171	1068,66	0,744
64	1678,28	0,968	172	1061,96	0,741
65	1673,14	0,967	173	1055,23	0,737
66	1667,99	0,966	174	1048,46	0,734
67	1662,84	0,965	175	1041,66	0,730
68	1657,69	0,964	176	1034,83	0,726
69	1652,52	0,963	177	1027,97	0,722
70	1647,36	0,962	178	1021,06	0,719
71	1642,18	0,961	179	1014,13	0,715
72	1637,00	0,960	180	1007,15	0,711
73	1631,82	0,958	181	1000,14	0,707
74	1626,63	0,957	182	993,08	0,703

k_tx (decimal) F0.8	Frequency (Hz)	Correction factor	k_tx (decimal) F0.8	Frequency (Hz)	Correction factor
75	1621,43	0,956	183	985,99	0,699
76	1616,22	0,955	184	978,86	0,695
77	1611,01	0,954	185	971,68	0,691
78	1605,79	0,952	186	964,47	0,687
79	1600,57	0,951	187	957,21	0,683
80	1595,33	0,950	188	949,90	0,679
81	1590,09	0,949	189	942,55	0,674
82	1584,85	0,947	190	935,15	0,670
83	1579,59	0,946	191	927,71	0,666
84	1574,33	0,945	192	920,21	0,661
85	1569,06	0,943	193	912,67	0,657
86	1563,79	0,942	194	905,07	0,652
87	1558,50	0,940	195	897,42	0,648
88	1553,21	0,939	196	889,72	0,643
89	1547,91	0,938	197	881,96	0,639
90	1542,60	0,936	198	874,14	0,634
91	1537,28	0,935	199	866,27	0,629
92	1531,96	0,933	200	858,33	0,624
93	1526,63	0,932	201	850,33	0,619
94	1521,28	0,930	202	842,27	0,614
95	1515,93	0,929	203	834,14	0,609
96	1510,57	0,927	204	825,94	0,604
97	1505,20	0,925	205	817,67	0,599
98	1499,82	0,924	206	809,33	0,594
99	1494,43	0,922	207	800,92	0,588
100	1489,04	0,921	208	792,42	0,583
101	1483,63	0,919	209	783,85	0,577
102	1478,21	0,917	210	775,20	0,572
103	1472,78	0,915	211	766,46	0,566
104	1467,34	0,914	212	757,63	0,561
105	1461,90	0,912	213	748,71	0,555
106	1456,44	0,910	214	739,70	0,549
107	1450,97	0,908	215	730,59	0,543
108	1445,49	0,907	216	721,37	0,537
109	1440,00	0,905	217	712,05	0,531
110	1434,49	0,903	218	702,62	0,524
111	1428,98	0,901	219	693,08	0,518
112	1423,46	0,899	220	683,41	0,511
113	1417,92	0,897	221	673,62	0,505
114	1412,37	0,895	222	663,70	0,498
115	1406,81	0,893	223	653,64	0,491
116	1401,24	0,891	224	643,44	0,484
117	1395,65	0,889	225	633,10	0,477
118	1390,05	0,887	226	622,59	0,470
119	1384,44	0,885	227	611,92	0,462
120	1378,82	0,883	228	601,07	0,455
121	1373,18	0,881	229	590,04	0,447
122	1367,53	0,879	230	578,81	0,439
123	1361,87	0,877	231	567,38	0,431
124	1356,19	0,875	232	555,73	0,423

k_tx (decimal) F0.8	Frequency (Hz)	Correction factor	k_tx (decimal) F0.8	Frequency (Hz)	Correction factor
125	1350,50	0,873	233	543,85	0,414
126	1344,79	0,870	234	531,71	0,406
127	1339,07	0,868	235	519,31	0,397
128	1333,33	0,866	236	506,63	0,387
129	1327,58	0,864	237	493,63	0,378
130	1321,82	0,861	238	480,31	0,368
131	1316,04	0,859	239	466,62	0,358
132	1310,24	0,857	240	452,54	0,348
133	1304,43	0,854	241	438,02	0,337
134	1298,60	0,852	242	423,03	0,326
135	1292,75	0,850	243	407,50	0,315
136	1286,89	0,847	244	391,39	0,303
137	1281,01	0,845	245	374,60	0,290
138	1275,11	0,842	246	357,05	0,277
139	1269,20	0,840	247	338,62	0,263
140	1263,27	0,837	248	319,14	0,248
141	1257,32	0,835	249	298,43	0,232
142	1251,35	0,832	250	276,21	0,215
143	1245,36	0,829	251	252,06	0,197
144	1239,36	0,827	252	225,37	0,176
145	1233,33	0,824	253	195,12	0,153
146	1227,29	0,821	254	159,26	0,125
147	1221,22	0,819	255	112,58	0,088

Table 0.1. Frequency values for tri-tones generator

amp (decimal) F8.0	Amplitude (dB)	amp (decimal) F8.0	Amplitude (dB)
1	-48,16	128	-6,02
2	-42,14	129	-5,95
3	-38,62	130	-5,88
4	-36,12	131	-5,81
5	-34,18	132	-5,75
6	-32,60	133	-5,68
7	-31,26	134	-5,62
8	-30,10	135	-5,55
9	-29,08	136	-5,49
10	-28,16	137	-5,43
11	-27,33	138	-5,36
12	-26,58	139	-5,30
13	-25,88	140	-5,24
14	-25,24	141	-5,18
15	-24,64	142	-5,11
16	-24,08	143	-5,05
17	-23,55	144	-4,99
18	-23,05	145	-4,93
19	-22,58	146	-4,87
20	-22,14	147	-4,81
21	-21,72	148	-4,75
22	-21,31	149	-4,70

amp (decimal) F8.0	Amplitude (dB)	amp (decimal) F8.0	Amplitude (dB)
23	-20,93	150	-4,64
24	-20,56	151	-4,58
25	-20,20	152	-4,52
26	-19,86	153	-4,47
27	-19,53	154	-4,41
28	-19,22	155	-4,35
29	-18,91	156	-4,30
30	-18,62	157	-4,24
31	-18,33	158	-4,19
32	-18,06	159	-4,13
33	-17,79	160	-4,08
34	-17,53	161	-4,02
35	-17,28	162	-3,97
36	-17,03	163	-3,92
37	-16,80	164	-3,86
38	-16,56	165	-3,81
39	-16,34	166	-3,76
40	-16,12	167	-3,71
41	-15,90	168	-3,65
42	-15,70	169	-3,60
43	-15,49	170	-3,55
44	-15,29	171	-3,50
45	-15,10	172	-3,45
46	-14,90	173	-3,40
47	-14,72	174	-3,35
48	-14,54	175	-3,30
49	-14,36	176	-3,25
50	-14,18	177	-3,20
51	-14,01	178	-3,15
52	-13,84	179	-3,10
53	-13,67	180	-3,05
54	-13,51	181	-3,01
55	-13,35	182	-2,96
56	-13,20	183	-2,91
57	-13,04	184	-2,86
58	-12,89	185	-2,82
59	-12,74	186	-2,77
60	-12,60	187	-2,72
61	-12,45	188	-2,68
62	-12,31	189	-2,63
63	-12,17	190	-2,58
64	-12,04	191	-2,54
65	-11,90	192	-2,49
66	-11,77	193	-2,45
67	-11,64	194	-2,40
68	-11,51	195	-2,36
69	-11,38	196	-2,31
70	-11,26	197	-2,27
71	-11,13	198	-2,23
72	-11,01	199	-2,18

amp (decimal) F8.0	Amplitude (dB)	amp (decimal) F8.0	Amplitude (dB)
73	-10,89	200	-2,14
74	-10,78	201	-2,10
75	-10,66	202	-2,05
76	-10,54	203	-2,01
77	-10,43	204	-1,97
78	-10,32	205	-1,92
79	-10,21	206	-1,88
80	-10,10	207	-1,84
81	-9,99	208	-1,80
82	-9,88	209	-1,76
83	-9,78	210	-1,72
84	-9,67	211	-1,67
85	-9,57	212	-1,63
86	-9,47	213	-1,59
87	-9,37	214	-1,55
88	-9,27	215	-1,51
89	-9,17	216	-1,47
90	-9,08	217	-1,43
91	-8,98	218	-1,39
92	-8,88	219	-1,35
93	-8,79	220	-1,31
94	-8,70	221	-1,27
95	-8,61	222	-1,23
96	-8,51	223	-1,19
97	-8,42	224	-1,16
98	-8,34	225	-1,12
99	-8,25	226	-1,08
100	-8,16	227	-1,04
101	-8,07	228	-1,00
102	-7,99	229	-0,96
103	-7,90	230	-0,93
104	-7,82	231	-0,89
105	-7,74	232	-0,85
106	-7,65	233	-0,81
107	-7,57	234	-0,78
108	-7,49	235	-0,74
109	-7,41	236	-0,70
110	-7,33	237	-0,67
111	-7,25	238	-0,63
112	-7,18	239	-0,59
113	-7,10	240	-0,56
114	-7,02	241	-0,52
115	-6,95	242	-0,48
116	-6,87	243	-0,45
117	-6,80	244	-0,41
118	-6,72	245	-0,38
119	-6,65	246	-0,34
120	-6,58	247	-0,31
121	-6,50	248	-0,27
122	-6,43	249	-0,24

amp (decimal) F8.0	Amplitude (dB)	amp (decimal) F8.0	Amplitude (dB)
123	-6,36	250	-0,20
124	-6,29	251	-0,17
125	-6,22	252	-0,13
126	-6,15	253	-0,10
127	-6,08	254	-0,06
		255	-0,03

Table 0.2. Amplitude values for tri-tones generator