TEXAS INSTRUMENTS

2.5G CS SW BU
ASD LOCOSTO

Nucleus N/A Locosto RN.n Architecture Specification Document

Document 13_04_09_01848 **Revision:** 1.0
Issue Date: 10 January, 2005

MakingWireless

__________ is a Trademark of Texas Instruments Incorporated

All other trademarks are the property of the respective owner.

Information in this document is subject to change without notice. Texas Instruments may have pending patent applications, trademarks, copyrights, or other intellectual property rights covering matter in this document. The furnishing of this document is given for usage with Texas Instruments products only and does not give you any license to the intellectual property that might be contained within this document. Texas Instruments makes no implied or expressed warranties in this document and is not responsible for the products based from this document.

Table of Contents

Revision History

Date	**Revision**	**Reason for Change**	**Changed By**	**Approval Authority**
10 Dec 2004	X0.1	Initial version of document.	Nicolas Estienne	
15 March-2005	0.2	Update after comments	Nicolas Estienne	
26 March 2005	1.0	Add Memory figures, features description	Nicolas Estienne	

REV	DATE	AUTHOR	NOTES
0.2	15 March 2005	Nicolas Estienne	Add critical use case, definition of SW components, list of SW components for ULC and Locosto+
1.0	26 March 2005	Nicolas Estienne	Add memory partitioning figures, main features description for all domains according to the PRD.

NOTES:

Note 1.

Plan Approvals

REV	APPROVAL 1	DATE	APPROVAL 2	DATE
0.2	Christophe Labaume		Eric Balard	
1.0	Christophe Labaume		Eric Balard	

NOTES: (DELETE this header and notes from your document)

Note 1. “Approval n” should be replaced by the approval authority (PT&R Team, SI Team, etc.)

Note 2. “Person” is the responsible signing representative for the authority.

Please read the “Important Notice” on the next page.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information 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 components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

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 or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information 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.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

1 Products		**2 Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.co	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265

1. Introduction

1.1. Purpose and Scope

This document is the top software architecture document of the Locosto product.

1.2. Product Overview

The scope is only on the architecture of the Software running on the ARM7 in the DBB. Software for DSP and DRP are out of the scope of this document. The functionalities like MP3, voice codec, radio burst processing which are provided by these parts are considered as given for the remaining architecture.

Studies and investigations took place in preparation to this document so that this document describes the target architecture and does not explain the way to come to it.

A maximum grade of re-use of SW which exists today in the TCS3.1 program should be achieved. Nevertheless the new HW capabilities and the new limitation of internal memory require changes which are intrusive in all main groups of the SW.

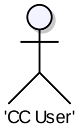
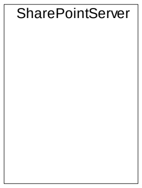
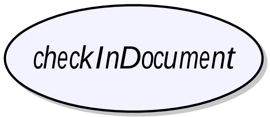
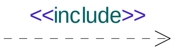
1.3. Limitations

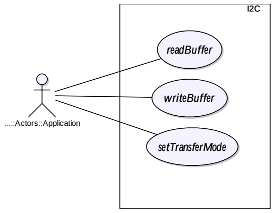
This document does not describe in detail the Locosto hardware platform. It is done the [SHI] document.

This document does not contain the DSP software, detailed L1 software.

1.4. UML Introduction

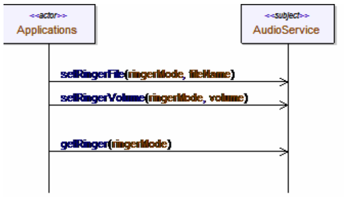
This document uses the UML 2.0 notation to describe the feature provided to the user and the dynamic description of this features. It uses symbols and concept described in the table below:

Symbol	Name	Description
	Actor	An actor is an end-user, software or a sub-system that plays a role in one or more interactions with your system.
	Subject	A subject represents a domain (a part of the system) involved in a use case.
	Use case	A use case symbol represents a function or a feature provided by a system or a sub-system
	<<include>>	The include relationship shows that this use case is call to perform the father use case.



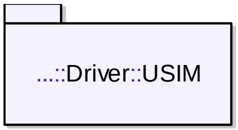
Use case diagram

A use case diagram is a graphical description of a feature that contains actors, subjects and use cases.


<<actor>>
Applications
<<subject>>
AudioService
setRingerFile(ringerMode, fileName)
setRingerVolume(ringerMode, volume)
getRinger(ringerMode)

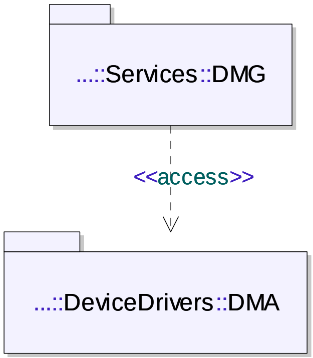
Sequence Diagram

A sequence diagram presents the flow in a high level view required to fulfill a feature. It is made of actor or subject (vertical line) that exchange message with or without parameters.



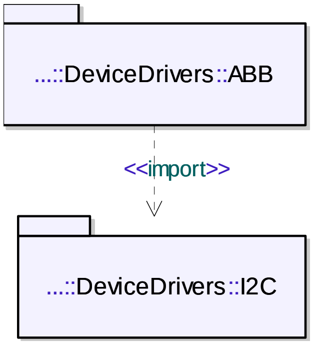
Package

A package is a logical and/or organizational view of software. A package should contain at least a software components or sub-packages.



<<access>> dependency

An <<access>> dependency shows that the DMG call DMA features using message and/or function call to perform its processing



<<import>> package dependencies

An <<import>> package dependency shows a strong dependency between two packages. It means that the package that imports an other can not perform its processing if the other package is not implemented.

2. Product Overview

2.1. Product Scope

The Locosto program is a new TI platform based on ARM7, a DSP and s DRP for the hardware and on maximum of re-use of the TCS 3.1 for the software.

The goal of the Locosto program is to provide a low cost chipset by internal memory limitation and increasing CPU clock in comparison with the Calypso+ chipset.

The main configuration defines for the Locosto platforms are:

- The **LocostoLite** provides a voice centric platform
  - the **Ultra Low Cost** platform (ULC) that contains no external SRAM (TCS2300)
  - The **LocostoLite Reference** that contains external RAM
- The **LocostoPlus** provides a GSM/GPRS platform with:
  - The **GPRS Basic** platform
  - The **GPRS standard** platform
  - The **GPRS enhanced**

2.2. Documentation tree

The ASD is the top level document of the software architecture. It gives a high level view of the architecture with the following information

- Main scope of the product
- Main features supported by the Locosto platform
- HW overview
- SW Architecture overview
- The list of SW components required

The detailed architecture is defined in the Domain architecture documents.

This initial version of the ASD shall be updated for the CP2 checkpoint and become the final ASD version.

2.3. Performance summary

This section describes CPU load, memory and power consumption study for the Locosto platform.

2.3.1. CPU Load

See sw_calculator.xls.

2.3.2. Memory footprint

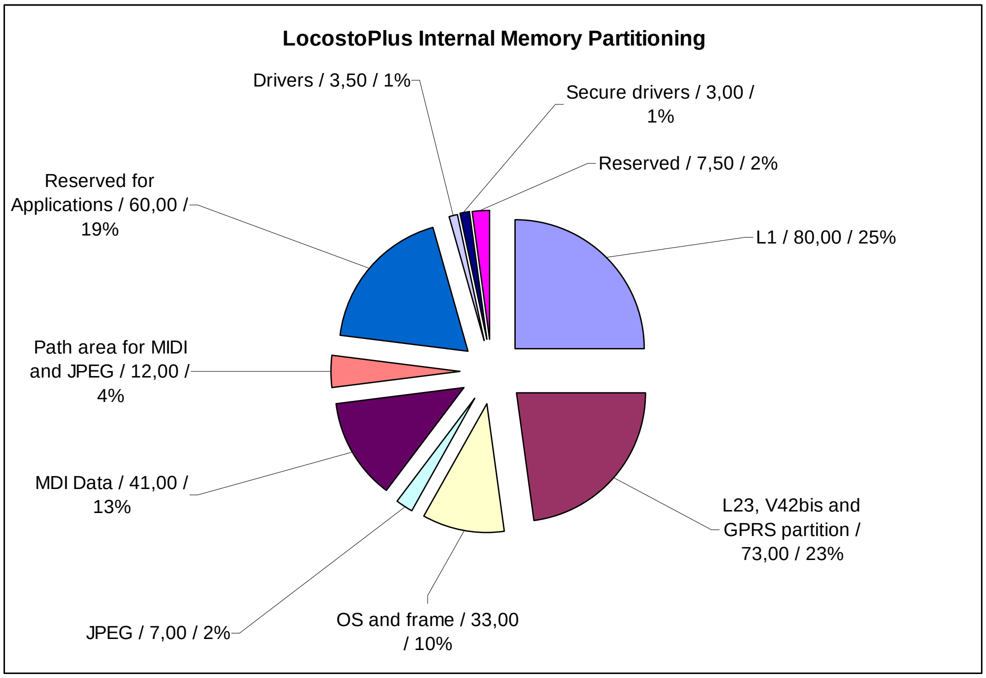
The table below describes the memory available according to the anchor definition.

	LocostoLite		**LocostoPlus**		
	TCS2300 (ULC)	TCS 2300-1 Reference	TCS2310	TCS2310-1	TCS2310-2
Embedded SRAM	2.5 Mbits				
External SRAM	None	2 Mbit	16 Mbits	16 Mbits	16 Mbits
External Flash	None	2 Mbytes	4 Mbytes	4 Mbytes	16 Mbytes

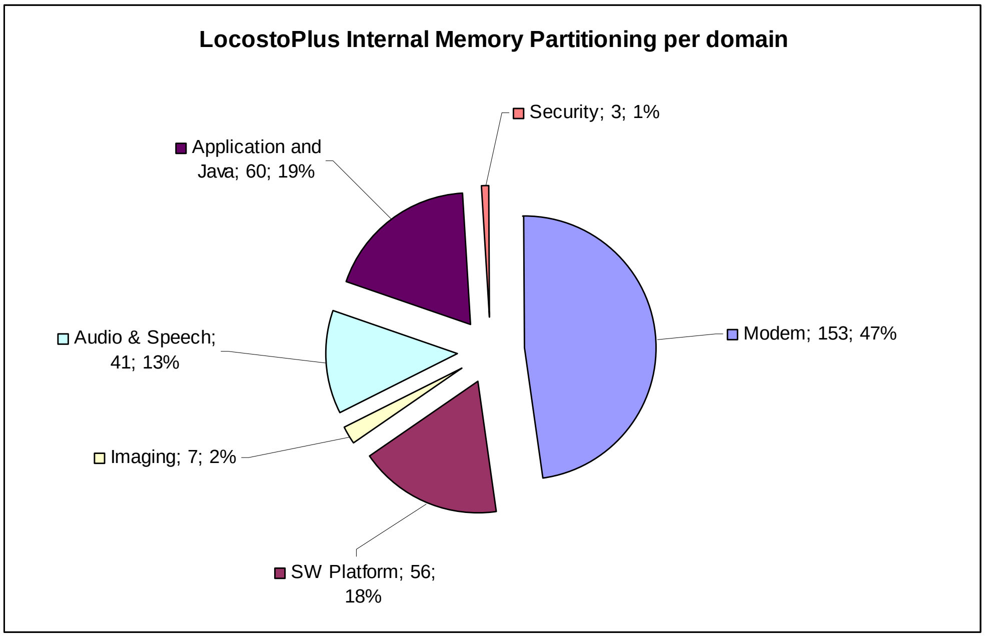
For the external flash, the maximum addressable space is set to 16 Mbytes.

2.3.2.1. Memory partitioning

The figure below presents the internal memory partitioning for the LocostoPlus configuration. All values are in Kbytes / percentages.

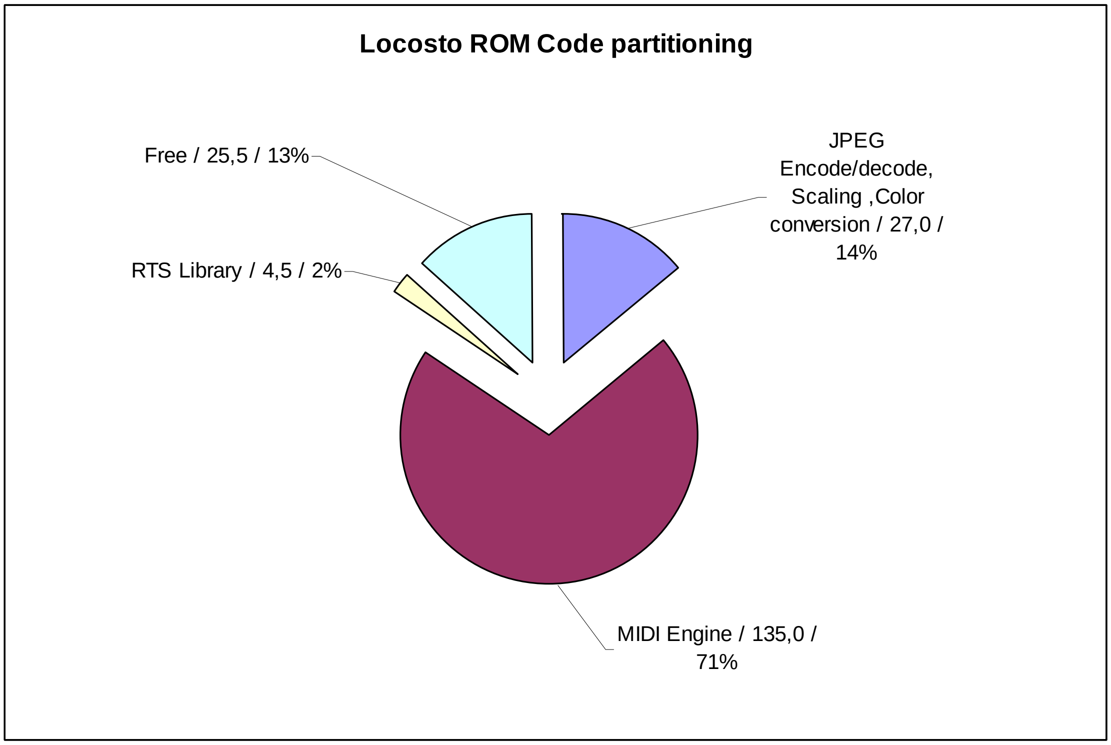


This figure gives the same values but filter by domain. All values are in Kbytes / percentages.



2.3.2.2. ROM Code

The figure below presents the ROM Code partitioning for all Locosto anchors.



2.3.2.3. Internal Memory

Executing code or accessing data from internal RAM has a direct impact on the CPU load since in this case code and data can be accessed with the full speed of the CPU. Running code from flash or accessing data located in the external RAM forces the CPU to introduce 9 wait-states at a speed of 104 MHz.

Since the internal RAM is a precious resource the trade-off between CPU load and internal RAM consumption is Key for the functional feasibility of LoCosto.

An analysis of the use cases regarding CPU load and fast memory requirements for code and data has been done in [7]. It is based on measurements and estimations and considers the anchors 2 to 4 only. Anchor 1 need to be analyzed separately because in this configuration the CPU requirements are not at the same order than in the full featured phones, so that the internal RAM should be used for data only in order to avoid the usage of external RAM.

The following content need to be mapped to the internal RAM:

For Anchor 1:

- All static data, stacks, partition pools
- No code can be run from internal RAM because of the memory limitation

For Anchors 2,3,4:

- User plane Partition Pools
- L1 code/data partly
- GRLC code/data partly
- RTS16 code/data partly
- Nucleus code/data partly (optional ROMed)
- V42bis code/data partly

The footprint for CPU load and internal RAM for the ARM7 running at 104Mhz looks as follow:

The colored fields are calculating the CPU load of the active parts of the use case. CPU load values are given in average value. Real time for CPU load means the required CPU load in order to fulfill the real time requirements of some blocks. The modem, the MIDI engine and the camera functionalities are real time dependent, whereas JAVA or a WAP browser could be delayed if no CPU is available. If the CPU load reaches 80% and above the use case is not feasible except if functionalities are removed or if a slowdown of non real-time parts could be accepted. The fixed CPU load should not exceed 80% in average.

To achieve the footprint the following optimizations need to be applied:

- The partition memory which is used as the dynamic memory allocation scheme needs to be split into two separate memory pools. One pool is located in the internal RAM and is only utilized for payload data in CSD and GPRS modem operations. The other pool is moved into the external RAM.
- The Beatnik MIDI BAE version 1.4 is split into code which requires fast execution and code which could run from flash. Today roughly 42k are identified which requires a fast execution. This code will be put in ROM on the LoCosto DBB and can be executed with 0ws with the full CPU speed. The fact that the ROM can not be changed after production the SW needs to be equipped with patch capabilities.
- The DLS which is in fact the sound bank used by the BAE for instrument synthesis requires also fast access to achieve a minimum in CPU load. This 100k DLS will be put also in ROM.
- JPEG codec will also be split into a time critical and non- time critical code and put into the ROM.
- It is optional to ROM time critical parts of Nucleus if required
- In order to share internal RAM space for data between JAVA, camera and BAE a dynamic memory management is required.
- Thanks to the capability of accessing external RAM from the DMA controller in burst mode the LCD frame-buffer is moved from internal memory to external RAM.

The following restrictions to the use cases are valid:

- In the use-cases #8 and #9 where JAVA, MIDI and CSD or JAVA, MIDI, GPRS, V.42bis and Bluetooth is running in parallel it is required to reduce MIDI to less than 8 voices mono in order to achieve the memory footprint. CPU load is close to the maximum of 80% which could result in slowdown of JAVA VM execution.
- In the use-cases #19 and #21 it is required to reduce the number of parallel MIDI voices to 24 in order to avoid CPU overloading.
- In general MIDI, Camera and Bluetooth and CSD/GPRS should not be used in parallel because the CPU load is out of bounds to the maximum load allowed.

2.3.2.4. Software partitioning

This section should contains partitioning figures for Locosto software.

2.3.3. Power consumption

See DAD power management.

2.4. Architecture overview

This section describes in detailed, the features supported by the Locosto anchors.

Modem features

The LocostoLite versions are GSM only mobile and the LocostoPlus are GSM/GPRS mobile. The Locosto platform does not support EDGE.

	LocostoLite		**LocostoPlus**		
	TCS2300 (ULC)	TCS 2300-1 Reference	TCS2310	TCS2310-1	TCS2310-2
GSM	Yes				
GPRS class 10	No		Yes		
Dual Band	Yes	No			
Quad Band	No	Yes			

Inconsistency in PRD: TCS2300-1: Dual or Quad

The Quad band configuration are: 2US+2EU or 2US+1EU or 1US+2EU.

Security features

All the Locosto anchors support EMEI protection, Flash content protection and SIM ME lock features.

	LocostoLite		**LocostoPlus**		
	TCS2300 (ULC)	TCS 2300-1 Reference	TCS2310	TCS2310-1	TCS2310-2
EMEI Protection	Yes				
Flash content protection	Yes				
SIM ME lock	Yes				

Graphics features

The standard LCD is a QCIF (144x176) form 4k to 64k color. Only the ULC should support a 94x94 black and white LCD.

<table>
<tr><th></th><th colspan="2">LocostoLite</th><th colspan="3">LocostoPlus</th></tr>
<tr><td></td><td>TCS2300
(ULC)</td><td>TCS 2300-1
Reference</td><td>TCS2310</td><td>TCS2310-1</td><td>TCS2310-2</td></tr>
<tr><td>LCD B&W (94x94)</td><td>Yes</td><td colspan="4">No</td></tr>
<tr><td>LCD Color QCIF (144x176) 4k color</td><td>No</td><td colspan="4">Yes</td></tr>
<tr><td>LCD Color QCIF 64k Color</td><td colspan="4">No</td><td>Yes</td></tr>
<tr><td>JPEG encode/decode</td><td colspan="2">No</td><td colspan="3">Yes</td></tr>
<tr><td>WBMP, aGIF, PNG</td><td colspan="5">Yes (*)</td></tr>
<tr><td>European Languagues</td><td colspan="4">Yes</td><td></td></tr>
<tr><td>Simplified Chinese</td><td colspan="4">Yes</td><td></td></tr>
</table>

Simplified Chinese language is exclusive with European languages.

(*) means delivered by the application suite.

Audio and Speech features

All Locosto anchors shall support MIDI play back and MID ringer features. The MP3 play back in ilde mode is targeted for the LocostoPlus.

	LocostoLite		**LocostoPlus**		
	TCS2300 (ULC)	TCS 2300-1 Reference	TCS2310	TCS2310-1	TCS2310-2
Voice codec: FR,HR, EFR	Yes				
Voice Codec: AMR	Yes				
Voice Memo	Yes				
MIDI Play back	16 tones	32 tones			
MIDI Ringer	Yes				
E1 player	Yes				
MP3 Play back in idle	No		Yes		
TTY	Yes				

The MIDI file formats supported are GM1, GM2, SP-MIDI, SMAF and XMF (Extended Midi File) with 100 Kbyte DLS and stereo.

For the ULC configuration, MIDI tone is limited to 16 voices in parallel, mono, 16bit/22kHz.

Imaging features

Only the LocostoPlus is able to support a camera sensor or the GoldenEye chip.

	LocostoLite		**LocostoPlus**		
	TCS2300 (ULC)	TCS 2300-1 Reference	TCS2310	TCS2310-1	TCS2310-2
Camera 300 KPixel	No		Yes		
Golden Eye	No				Yes

Inconsistency in PRD: See CAMERA in table

The Camera should provides VGA quality pictures and shutter noise.

Application and Java features

The following application shall be supported on the Locosto platform but all are not delivered in the TCS 3.2. (*) means not delivered.

	LocostoLite		**LocostoPlus**		
	TCS2300 (ULC)	TCS 2300-1 Reference	TCS2310	TCS2310-1	TCS2310-2
SMS	Yes				
EMS (*)	No		Yes		
MMS (*)	No		Yes		
Customizable messaging	(?)		(?)		
JAVA MIDP 2.0 (*)	No		Yes		
WAP Browser (*)	No		Yes		
WAP Browser Dual stack (*)	No		Yes		
Native Games (*)	No		Yes		
Predictive Input (*)	No	Yes			

The following JSR should be supported by the JVM running on Locosto:

- JSR 120 (WMA)
- JSR 135 (MMA)
- JSR 177 (Security)

Connectivity features

	LocostoLite		**LocostoPlus**		
	TCS2300 (ULC)	TCS 2300-1 Reference	TCS2310	TCS2310-1	TCS2310-2
USB Client 1.1 Modem	No (?)		Yes		
USB Trace	No(?)		Yes		
USB Client 1.1 / Charger	Yes				
IrDA	No		Yes		
BlueTooth	No				Yes

The Bluetooth profiles supported are:

- Headset profile
- PC Dial-up support profile
- Mass storage profile

Peripheral features

	LocostoLite		**LocostoPlus**		
	TCS2300 (ULC)	TCS 2300-1 Reference	TCS2310	TCS2310-1	TCS2310-2
Vibrator	Yes				
Buzzer	Yes, configurable				
Hands free	No	Yes			
Headset	No	Yes			
Car Kit	No	Yes			

Test and tools features

	LocostoLite		LocostoPlus		
	TCS2300 (ULC)	TCS 2300-1 Reference	TCS2310	TCS2310-1	TCS2310-2
Test MMI	Yes				

The test MMI should provide features to demonstrate Call management, MIDI, MP3 and Camera features.

The software will be configurable via the build system. Feature flags are used to compile for different configurations.

Limitations for ULC Configuration (TCS 2300)

- The footprint for this anchor can only be achieved by applying the following limitations:
- No code is executed from internal RAM
- All time-critical parts of the MIDI BEA 1.4_optimized is located and executed from internal ROM of LoCosto DBB
- A serialized 100k DLS is located and accessed from internal ROM
- As LCD a 160x120 pixel 4k color or B&W 96x96 pixel 4k is supported (18k Frambuffer)
- L1, L23, Platform (Drivers, Services, RTOS, RTS, Startup), MFW need to achieve a footprint of 1,2MByte Flash and 166 kByte internal RAM.
- A dynamic download of MIDI ring-tones with up to 2k content is supported only
- ~~No USB modem support,, no external/internal string based AT command interface supported.~~
- Flash download and trace only via USB
- 85k internal RAM and 800k Flash is reserved for customer MMI (including fonts, icons, EU language, simpl. chinese + T9). If the customer MMI is out of this margin a larger flash memory (4Mbit) need to be used instead.
- Margin of 20k internal RAM kept in order to minimize risk of unknown MMI requirements.

2.5. Critical use cases

This section contains the list of critical use cases detected that have impact on performance and resource. It represents the worst case in term of CPU load requirements

UC ReqID	Description	Feature ReqID
	1 - Accept mobile terminated voice call, phone is in idle state	
	2 - Voice memo record/play while voice call	
	3 – SMS Reception/MMS notification while in dedicated mode	
	4 – Voice call MO/MT with Bluetooth connection	
	5 – Upload/download MMS via CSD	
	6 - Upload/download MMS via GPRS	
	7 – Java application running while voice call via Bluetooth	
	8 - Java application running while MIDI sound and CSD data connection	
	9 - Java application running while MIDI sound GPRS data connection with Bluetooth	
	10 - Java application running while voice call via Bluetooth	
	11 – Camera view finder in idle mode	
	12 - Camera snapshot in idle mode	
	13 - Camera view finder while voice call	
	14 - Camera snapshot while voice call	
	15 - Camera view finder while GPRS data transfer	
	16 - Camera snapshot while GPRS data transfer	
	17 – Incoming call with MIDI ringer tone while camera view finder mode	
	18 - Incoming call with MIDI while camera snapshot mode	
	19 - Incoming call with MIDI while Java application is executed with MIDI sound	

20 - Incoming call with MIDI playback via Bluetooth headset connection while idle mode

21 - Incoming call with MIDI playback via Bluetooth headset connection while idle mode

22 – Java network game application with MIDI with Headset: PC synchronization via USB

23 – Voice call with TTY

24 - MIDI playback during background GPRS data transfer and alert generation for incoming voice call

25 - AMR-NB playback during background GPRS data transfer and alert generation for incoming voice call

26 - Camera View finder with MIDI player

27 - Camera Snap-shot with MIDI player (with 2.5S Shot-2-Shot delay)

Figure 1 : Critical Use cases

3. Software Architecture

This section describes the SW architecture of the Locosto platform. It means:

- Identify all software component that are going to develop or reuse from TCS 3.1 to fulfill the TCS 3.2 requirements
- Provide a short description for each
- Draw the dependencies between this SW component

3.1. SW architecture overview

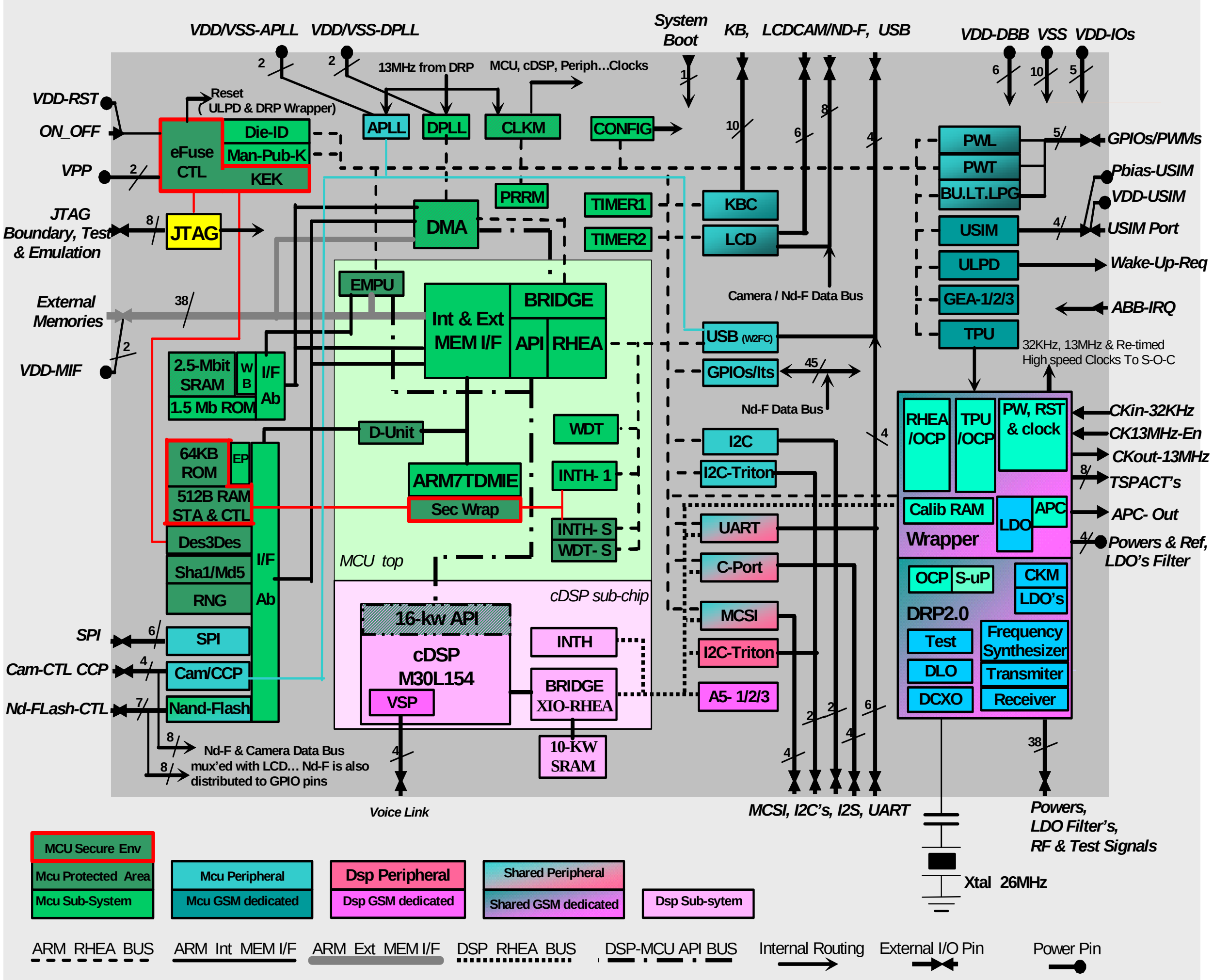
3.1.1. Hardware architecture overview

A detailed description of the blocks can be found in the LoCosto System On Chip Specification [2].

The DBB contains:

- ARM7 @104Mhz with access to 2.5Mbit internal SRAM and 1.5Mbit ROM
- cDSP M30LO154 @104Mhz with 10KW SRAM and 16KW API RAM
- DRP 2.0 with wrapper
- Crypto units with eFuse and keys
- DMA controller
- HW Timer
- IRQ control
- 64K Secure ROM
- JTAG
- Memory interface to external SRAM and Flash memory
- HW controller for 1xUSB, 1x1UART, SPI, LCD, Camera, USIM, 2xI2C, NAND-Flash, I2S, MCSI, Keypad, LCD etc.

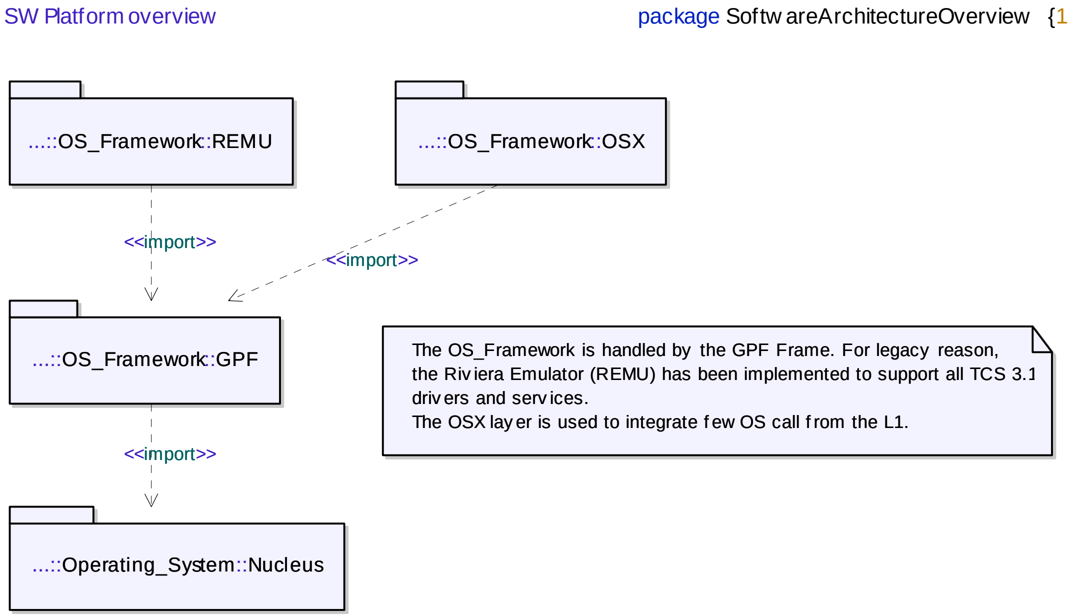
In the following picture the functional blocks and peripherals are shown for the DBB LocCosto-IC.


VDD/VSS-APLL
VDD/VSS-DPLL
System Boot
KB, LCDCAM/ND-F, USB
VDD-DBB VSS VDD-IOs
13MHz from DRP
MCU, cDSP, Periph...Clocks
Reset
(ULPD & DRP Wrapper)
VDD-RST
ON_OFF
VPP
JTAG Boundary, Test & Emulation
External Memories
VDD-MIF
eFuse CTL
Die-ID
Man-Pub-K
KEK
JTAG
APLL
DPLL
CLKM
CONFIG
PRRM
TIMER1
TIMER2
DMA
KBC
LCD
EMPU
Int & Ext MEM I/F
BRIDGE
API
RHEA
2.5-Mbit SRAM
WB
I/F Ab
1.5 Mb ROM
D-Unit
WDT
ARM7TDMIE
Sec Wrap
INTH- 1
INTH- S
WDT- S
MCU top
64KB ROM
EP
512B RAM STA & CTL
Des3Des
Sha1/Md5
RNG
I/F Ab
SPI
Cam/CCP
Nand-Flash
cDSP sub-chip
16-kw API
cDSP M30L154
VSP
INTH
BRIDGE XIO-RHEA
10-KW SRAM
Camera / Nd-F Data Bus
USB (W2FC)
GPIOs/Its
Nd-F Data Bus
I2C
I2C-Triton
UART
C-Port
MCSI
I2C-Triton
A5- 1/2/3
PWL
PWT
BU.LT.LPG
USIM
ULPD
GEA-1/2/3
TPU
GPIOs/PWMs
Pbias-USIM
VDD-USIM
USIM Port
Wake-Up-Req
ABB-IRQ
32KHz, 13MHz & Re-timed High speed Clocks To S-O-C
RHEA /OCP
TPU /OCP
PW, RST & clock
Calib RAM
LDO
APC
Wrapper
OCP
S-uP
CKM
LDO's
DRP2.0
Test
DLO
DCXO
Frequency Synthesizer
Transmiter
Receiver
CKin-32KHz
CK13MHz-En
CKout-13MHz
TSPACT's
APC- Out
Powers & Ref, LDO's Filter
SPI
Cam-CTL CCP
Nd-FLash-CTL
Nd-F & Camera Data Bus mux'ed with LCD... Nd-F is also distributed to GPIO pins
Voice Link
MCSI, I2C's, I2S, UART
Powers, LDO Filter's, RF & Test Signals
Xtal 26MHz
2
2
1
6
10
5
10
6
8
4
5
4
8
38
2
2
4
45
4
8
6
4
7
8
8
4
4
2
2
4
6
38
MCU Secure Env
Mcu Protected Area
Mcu Sub-System
Mcu Peripheral
Mcu GSM dedicated
Dsp Peripheral
Dsp GSM dedicated
Shared Peripheral
Shared GSM dedicated
Dsp Sub-sytem
ARM RHEA BUS
ARM Int MEM I/F
ARM Ext MEM I/F
DSP RHEA BUS
DSP-MCU API BUS
Internal Routing
External I/O Pin
Power Pin

3.1.2. Software Architecture overview

This section contains the package diagram of each domain of the Locosto. The OS_Framework is handled by the GPF Frame. For legacy reason, the Riviera Emulator (REMU) has been implemented to support all TCS 3.1 drivers and services. The OSX layer is used to integrate few OS call from the L1. The Modem contains: - the Modem Interface - AT Command Interface (ACI) - Layer 23 - Layer 1

This section contains the package diagram of each domain of the Locosto. The OS_Framework is handled by the GPF Frame. For legacy reason, the Riviera Emulator (REMU) has been implemented to support all TCS 3.1 drivers and services. The OSX layer is used to integrate few OS call from the L1. The Modem contains: - the Modem Interface - AT Command Interface (ACI) - Layer 23 - Layer 1



The OS_Framework is handled by the GPF Frame. For legacy reason, the Riviera Emulator (REMU) has been implemented to support all TCS 3.1 drivers and services.
The OSX layer is used to integrate few OS call from the L1.

Figure 2. SW Platform overview (Package Diagram)

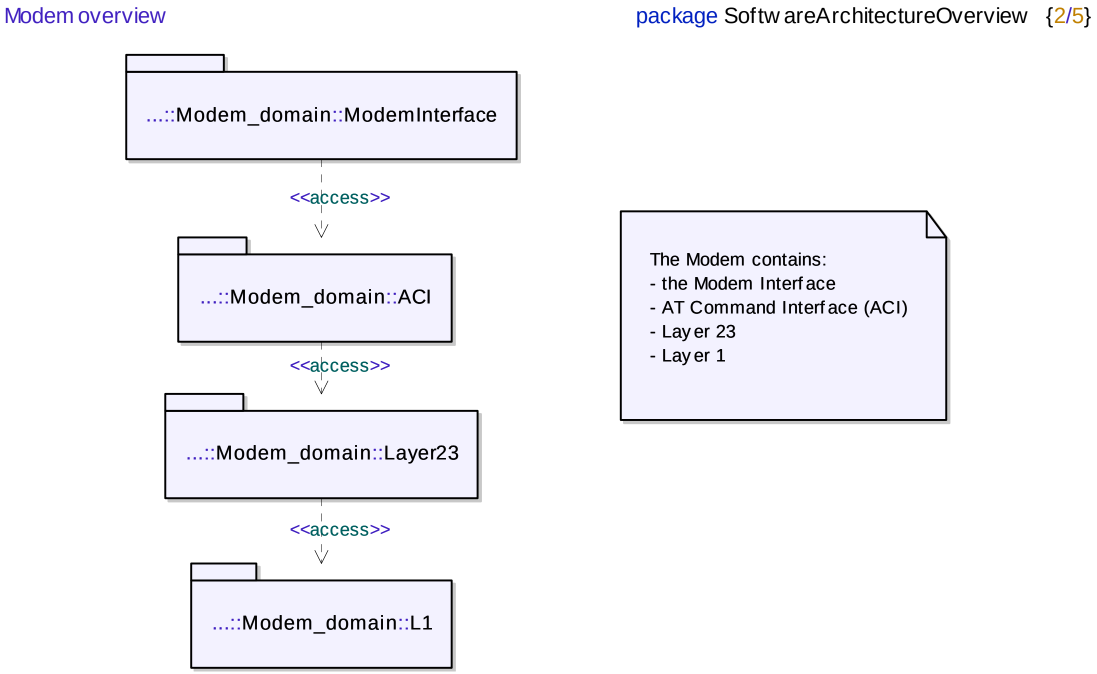


Figure 3. Modem overview (Package Diagram)

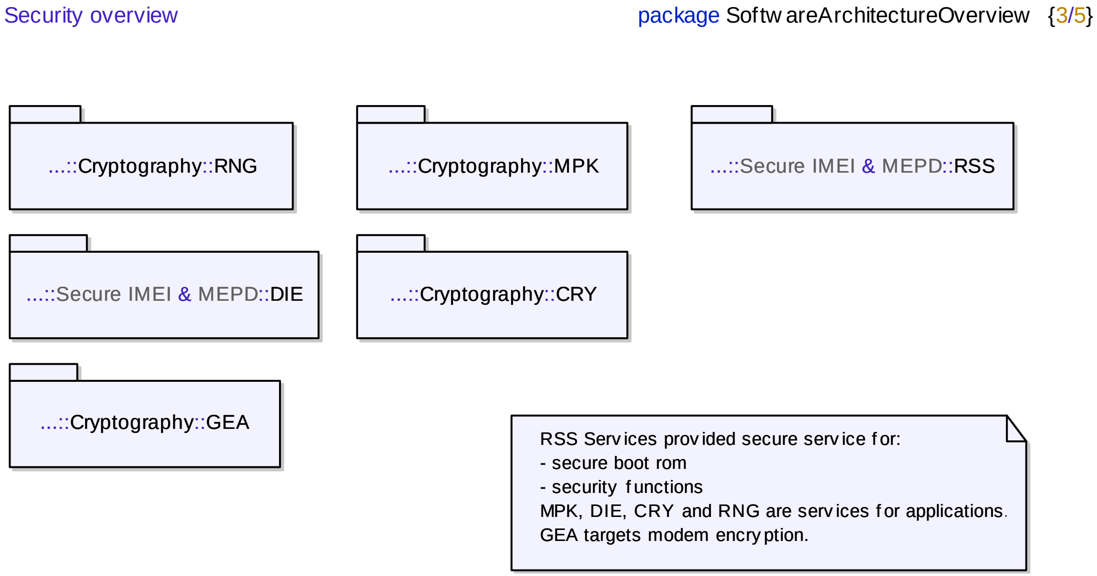

Security overview
package Softw areArchitectureOverview {3/5}
...::Cryptography::RNG
...::Cryptography::MPK
...::Secure IMEI & MEPD::RSS
...::Secure IMEI & MEPD::DIE
...::Cryptography::CRY
...::Cryptography::GEA
RSS Serv ices prov ided secure serv ice f or:
- secure boot rom
- security f unctions
MPK, DIE, CRY and RNG are serv ices f or applications.
GEA targets modem encry ption.

Figure 4. Security overview (Package Diagram)

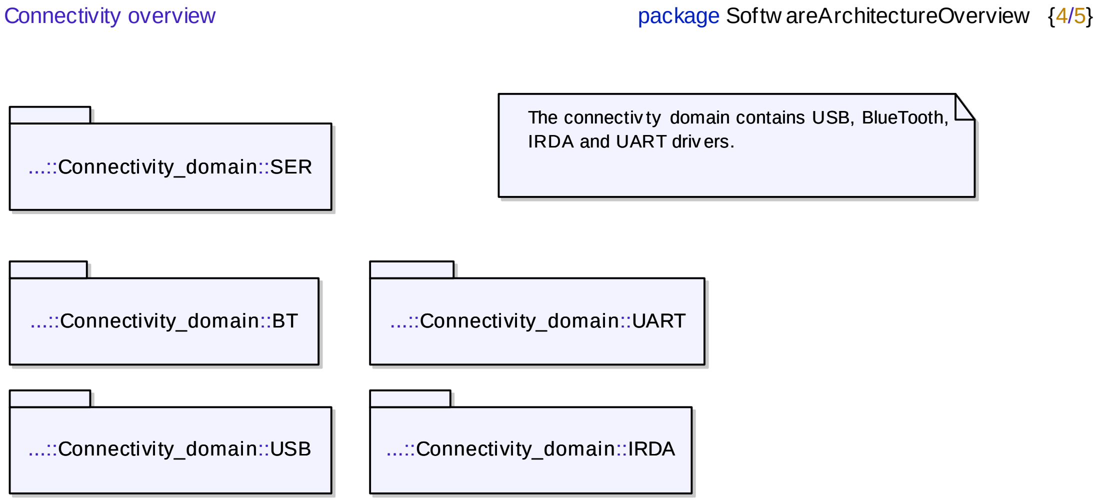

Connectivity overview
package SoftwareArchitectureOverview {4/5}
...::Connectivity_domain::SER
The connectivty domain contains USB, BlueTooth, IRDA and UART drivers.
...::Connectivity_domain::BT
...::Connectivity_domain::UART
...::Connectivity_domain::USB
...::Connectivity_domain::IRDA

Figure 5. Connectivity overview (Package Diagram)

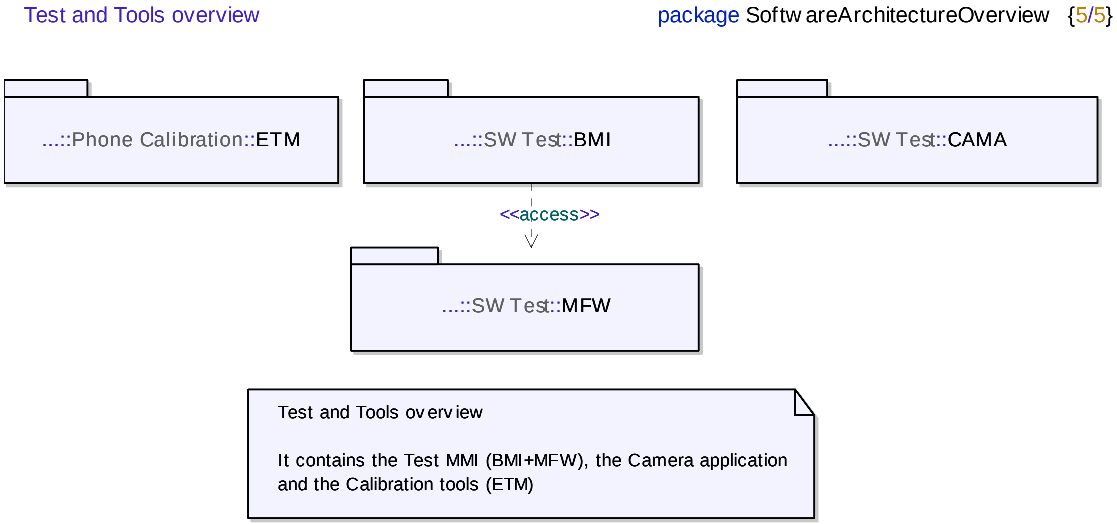

Test and Tools overview
package SoftwareArchitectureOverview {5/5}
...::Phone Calibration::ETM
...::SW Test::BMI
...::SW Test::CAMA
<<access>>
...::SW Test::MFW
Test and Tools overview
It contains the Test MMI (BMI+MFW), the Camera application and the Calibration tools (ETM)

Figure 6. Test and Tools overview (Package Diagram)

3.2. Component description

This section gives a static view of the SW components involved in all anchors of the Locosto program.

3.2.1. OS_and_SW_Platform_domain

Purpose

The Operating System and Software Platform (OS_SWP) domain contains all the basic software and the platform.

Main functions

The Nucleus OS is a Real Time Operating System providing all basic features of an operating system. Its main advantages are:

- a portable OS cross over ARM architecture
- pre-emptive multi tasking
- tasks synchronization
- management of interruption

In order to give abstract views of the OS, three frameworks are running:

- The Generic Protocol stack Frame (GPF) providing functions used by the GSM/GPRS Layer 2/3. This frame is more modem feature oriented.
- The Cross Operation System (OXS) providing functions used by the Layer 1 (L1) software. It is based on GPF functionalities.
- The Riviera Framework (RVF) used to manage Software Entity. This frame is more application features oriented. The RVF is now based on GPF functionalities.

This package dependencies diagram that OSX and RVF packages are based on GPF features. Some features of RVF that not exist in GFP use directly Nucleus functions.

Restrictions

None.

OS and SW Platform dependencies

All Locosto software are based on the OS and
SW Platform domain.

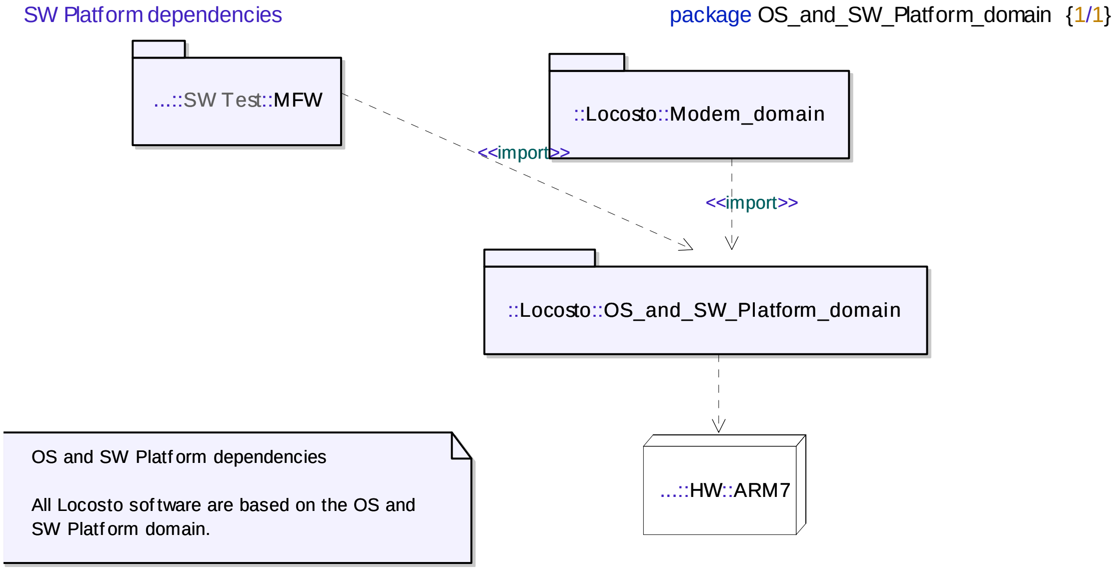

SW Platform dependencies
package OS_and_SW_Platform_domain {1/1}
...::SW Test::MFW
::Locosto::Modem_domain
<<import>>
<<import>>
::Locosto::OS_and_SW_Platform_domain
...::HW::ARM7
OS and SW Platform dependencies
All Locosto software are based on the OS and SW Platform domain.

Figure 7. SW Platform dependencies (Package Diagram)

3.2.1.1. Operating_System

3.2.1.1.1. NUCLEUS

Purpose

The Nucleus package contains the RTOS running on Locosto platform.The LoCosto platform uses the 1.0.G1.0f. version.

Main functions

Nucleus from Accelerated Technology is used as RTOS. Nucleus provides functionalities for multithreaded execution of code with shared text and data segment. For handling of interrupts low level functions LISR and high level functions HISR are available to implement a preemptive and prioritized execution of interrupt service code. Threads use, HISR and LISR may use separate execution stacks. Nucleus also provides functionalities to implement Mutex operations, Timers and dynamic/partition oriented memory allocation schemes. For Inter Process Communication several function are provided.

Some parts of the Nucleus code are going to be ROM in order to reduce the internal RAM footprint and keep the advantage of code execution with no access time penalty.

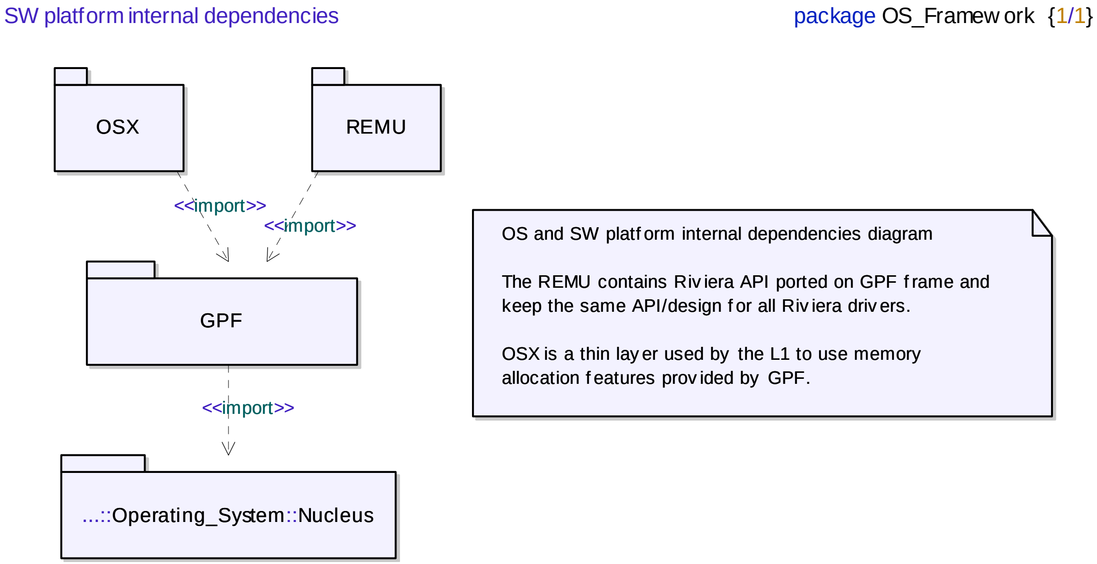
Restrictions

3.2.1.2. OS_Framework

OS and SW platform internal dependencies diagram

The REMU contains Riviera API ported on GPF frame and keep the same API/design for all Riviera drivers.

OSX is a thin layer used by the L1 to use memory allocation features provided by GPF.


SW platform internal dependencies
package OS_Framework {1/1}
OSX
REMU
<<import>>
<<import>>
GPF
<<import>>
...::Operating_System::Nucleus

OS and SW platform internal dependencies diagram

The REMU contains Riviera API ported on GPF frame and keep the same API/design for all Riviera drivers.

OSX is a thin layer used by the L1 to use memory allocation features provided by GPF.

Figure 8. SW platform internal dependencies (Package Diagram)

3.2.1.2.1. GPF

Purpose

The Generic Protocol Stack Framework (GPF) provides a framework for all software entity running on the platform.

Main Functions

- Task Management
- Memory Management
- Communication
- Timer Management
- Routing
- Tracing
- Test Interface
- RTOS Access
- Main loops of processes (passive body)

Restriction

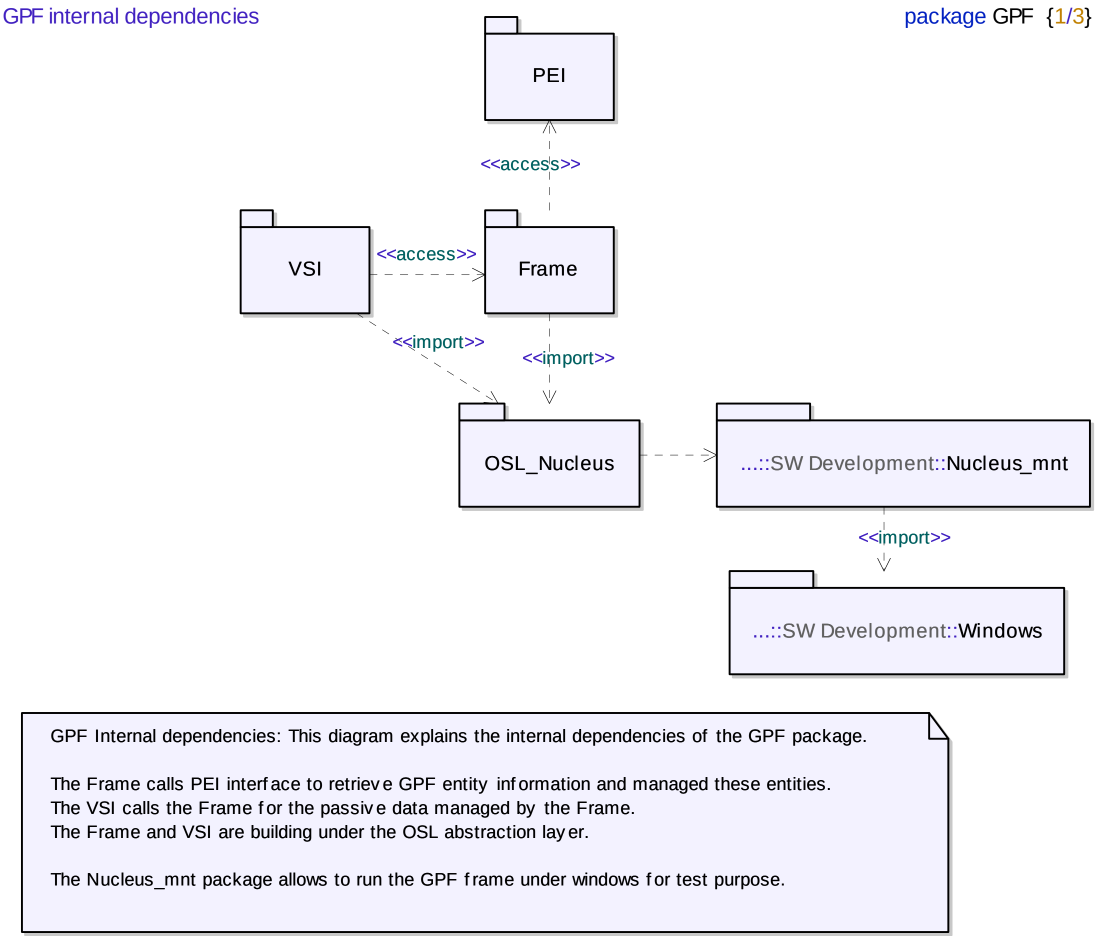
GPF dependencies diagram

This diagram shows the dependencies of GPF with external pacakges
- OSL_Nucleus uses DAR services to store warning and error messages for recovery, and to reset in case of fatal error.
- The VSI access to FFS in order to read the trace filter.

GPF Internal dependencies: This diagram explains the internal dependencies of the GPF package.

The Frame calls PEI interface to retrieve GPF entity information and managed these entities.
The VSI calls the Frame for the passive data managed by the Frame.
The Frame and VSI are building under the OSL abstraction layer.

The Nucleus_mnt package allows to run the GPF frame under windows for test purpose.

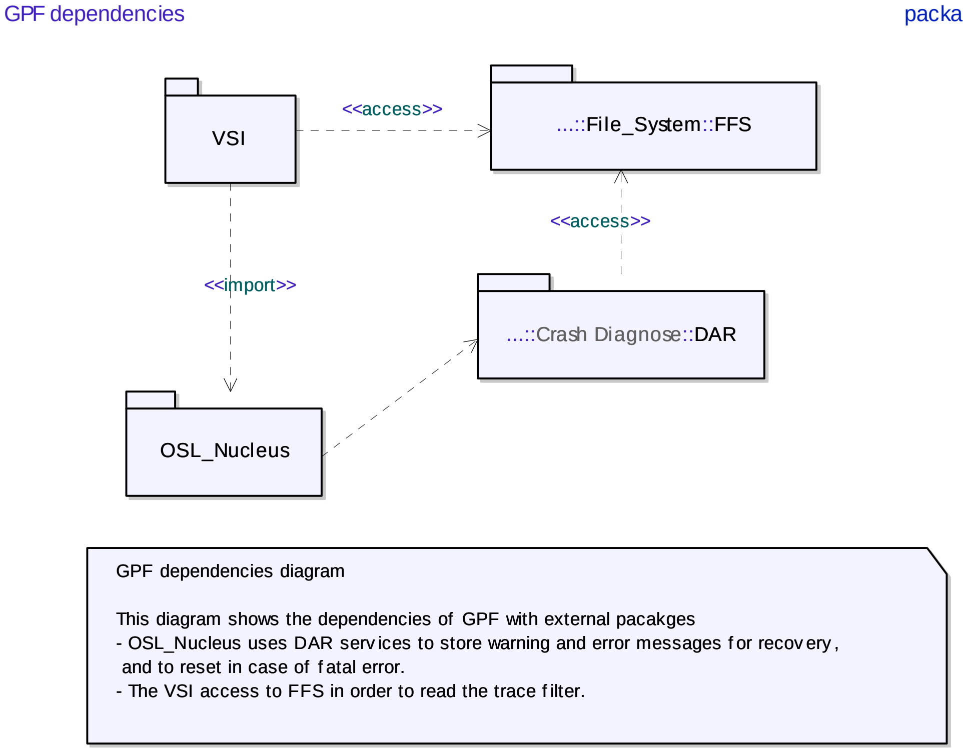


GPF Internal dependencies: This diagram explains the internal dependencies of the GPF package.

The Frame calls PEI interface to retrieve GPF entity information and managed these entities.
The VSI calls the Frame for the passive data managed by the Frame.
The Frame and VSI are building under the OSL abstraction layer.

The Nucleus_mnt package allows to run the GPF frame under windows for test purpose.

Figure 9. GPF internal dependencies (Package Diagram)



GPF dependencies diagram

This diagram shows the dependencies of GPF with external pacakges
- OSL_Nucleus uses DAR services to store warning and error messages for recovery, and to reset in case of fatal error.
- The VSI access to FFS in order to read the trace filter.

Figure 10. GPF dependencies (Package Diagram)

3.2.1.2.1.1. PEI

Purpose

The Protocol Entity Interface (PEI) is the functional interface of the protocol entities to be accessed by the frame. Any entity running under GPF frame should implement this interface.

Mains functions

A protocol stack entity is registered to the system by calling the pei_create() function that exports the user defined entity creation parameters.

The communication between the protocol stack entities is done via message queues. These message queues are created by the frame during the system startup and the communication between the entities is opened by the function pei_init(). A message identifier defines the type of the received queue entry. Three different kinds of entries can be transferred through these queues: primitives, signals and timeouts. Each of these message types has a corresponding PEI f unction that is called if a message has been received.

Setting of dynamic configurations of protocol stack entities e.g. timer configuration as well as reading of these configurations is performed by the function pei_config().

Restrictions

3.2.1.2.1.2. VSI

Purpose

The Virtual System Interface (VSI) is the functional interface of the frame seen by the body. For the bodies, the VSI serves as an interface to the underlying OS Layer that is an abstraction of the RTOS (Nucleus, Win32,...). The RTOS resources created are accessed through the VSI that calls OS Layer functions, some of them e.g. the timers are managed in the VSI. T he VSI also provides access to extended frame functionality such as tracing and memory supervision.

Description VSI: Virtual System Interface
+Task Management
+ Memory
  - allocate, free, get status, attach to memory partition
  - Allocate root of dynamic primitves, root of dynamic memory, additional dynamic memory
  - get/free dynamic sized memory

+ COM
+ Software Timer
  - start/stop timer
  - start timer with periodic reload
  - get timer status
  - configure timer
  - get system clock
  - suspend thread (?)
+ Routing
+ Tracing
+ Test
+ Semaphore management
  - Open/close/get/release Semaphore
  - query semaphore counter value

- Mains loops of processes: active body in VSI and passive body in the Frame.

3.2.1.2.1.3. Frame

Purpose

The Frame package is the passive body part of the GPF frame. It manages all entity information required to manage task.

Main functions

The unique function provided by the Frame is the start frame function. This function performs the following operations:
- Read configuration
- Create and start all the tasks
- Provide task entry functions
- Create message queues
- Contains main loop for passive body.

Restrictions

3.2.1.2.1.4. OSL_Nucleus

Purpose

This OS layer handle the abstraction layer of the operating system. It allows to be independent to the RTOS.

3.2.1.2.1.5. TST

Purpose

The Test Software entity allows to transfer trace for the entity to the tools side.

Main function

This TST SW provides to functions
- open driver for initialization
- write to write trace

Restrictions

SW components

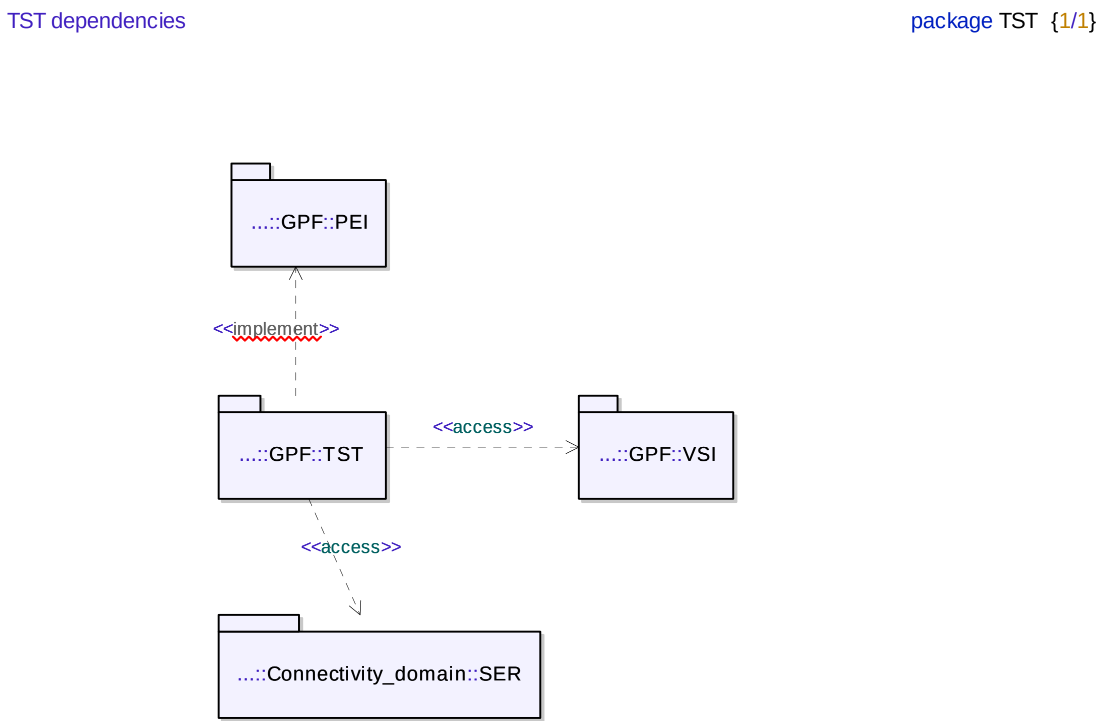


Figure 11. TST dependencies (Package Diagram)

3.2.1.2.2. REMU

Purpose

The Riviera Emulator (REMU) allows to run all Riviera drivers in a GPF task. The legacy Riviera Frame (RVF) and the Riviera Manager (RVM) are the two main components that provide facilities for task management, memory mechanism and communication between components.

For Locosto platform, the Riviera drivers:
- expose the same API as the TCS 3.1
- use the same RVM/RVF API.

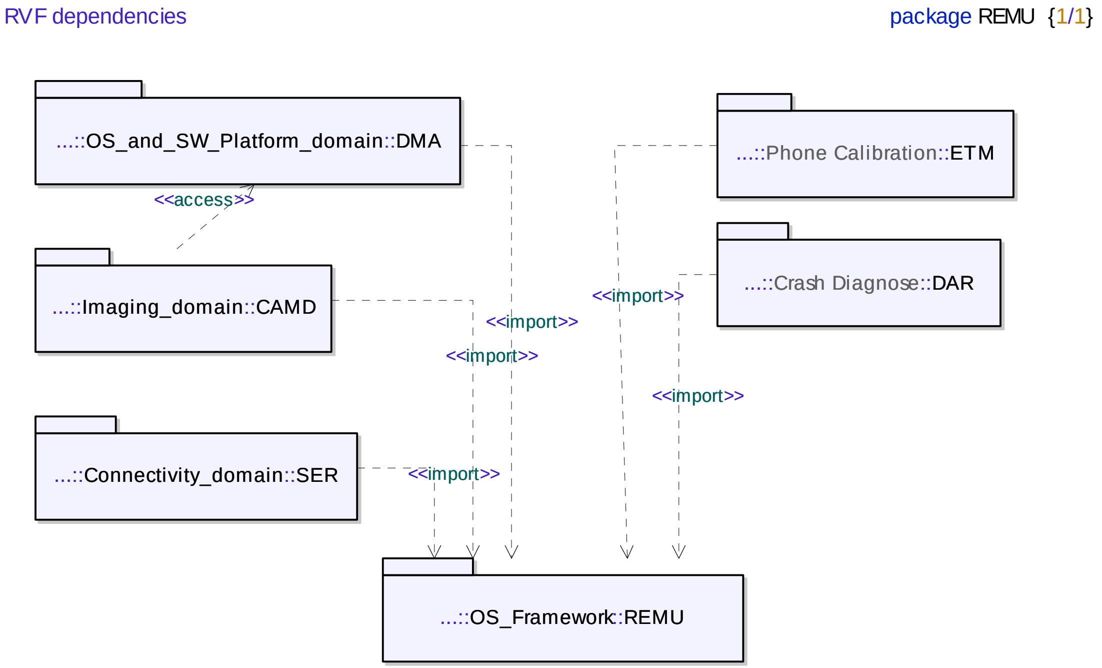


Figure 12. RVF dependencies (Package Diagram)

3.2.1.2.3. OSX

Purpose

The OSX interface is a thin layer that allows to use GPF functions in the Layer 1 package.

Main Function

It provides the following functions to L1 packages:
- allocate/deallocate memory partition
- allocate/deallocate of memory buffer
- send/receive message
- send signal
- initialize/configuration
- open queue

Restriction

3.2.1.3. Hardware_Configuration

3.2.1.3.1. FPM

Purpose

The Functional Pin Muxing (FPM) allows to configure PIN for exclusive functional use in runtime mode.

Main functions

To be completed.

Restrictions

Static initiliazation. not a real driver.

3.2.1.3.2. UICC

Purpose

The purpose of the SIM Driver is to provide a low-level access to the SIM Card content. The SIM Driver API is based on the ETSI/3GPP 11.11 and ISO/CEI 7816-3 commands.

Main functions

The SIM functions sit in five sets of functions:
- Initialize, clean-up
o register callbacks for insertion and removal of the card
o initialization of the SIM driver and timer
o reset the SIM card that means "hardware" initialization of the card
o un-register SIM card (shutdown signals, clean the registers and delete timer)
- Card holder verification
o verify/change the Card Holder Verification (CHV)
o enable/disable the CHV
o unblock the CHV
- General operations
o read SIM status/read extended SIM status
o get response
- File management
o perform SELECT on card
o seek pointer to a position on the card
o read/update record on card
o read/update binary content on the card
o move pointer to specific record number of the card
- Special operations
o execute the a3-A8 algorithm
o invalidate/rehabilitate a file
- Toolkit operations
o read capabilities of the mobile
o perform FETCH command on card
o transfer response to SIM after FETCH command (TERMINAL RESPONSE command)
o send data to the SIM (ENVELOPE command)

Restrictions

The SIM driver doest not handle Layer2/Layer3 SIM card access functions.

Reuse USIM Neptune driver.
USIM dependencies

The USIM driver is only used by the Layer23.

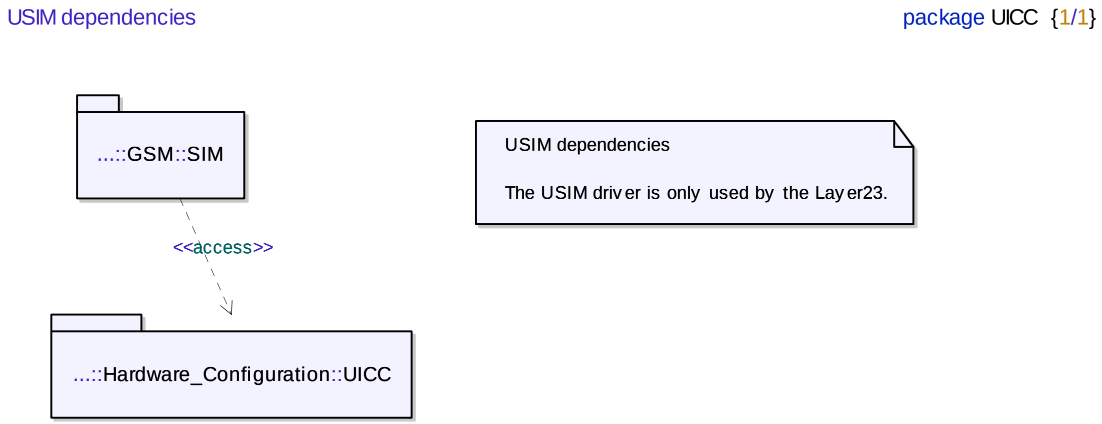

USIM dependencies
package UICC {1/1}
...::GSM::SIM
<<access>>
...::Hardware_Configuration::UICC
USIM dependencies
The USIM driver is only used by the Layer23.

Figure 13. USIM dependencies (Package Diagram)

3.2.1.4. Clock_and_Alarm

3.2.1.4.1. RTC

Purpose
The Real Time Clock (RTC) driver handles date/time information on the system, through its interface with the dedicated RTC hardware running at 32 kHz, independently from the CPU.

Main functions
The RTC driver allows to:
- get and set current date and time,
- get and set alarm date and time,
- activate and deactivate alarm,
- round the internal clock to the closest minute (30 second time correction),
- alert any registered SW entity when date and time alarm are reached

Restrictions

Rework: because RTC Triton manages the RTC.
See dependencies with PS ?

RTC dependencies diagram

The RTC driver uses I2C driver to access to ABB triton clock features.

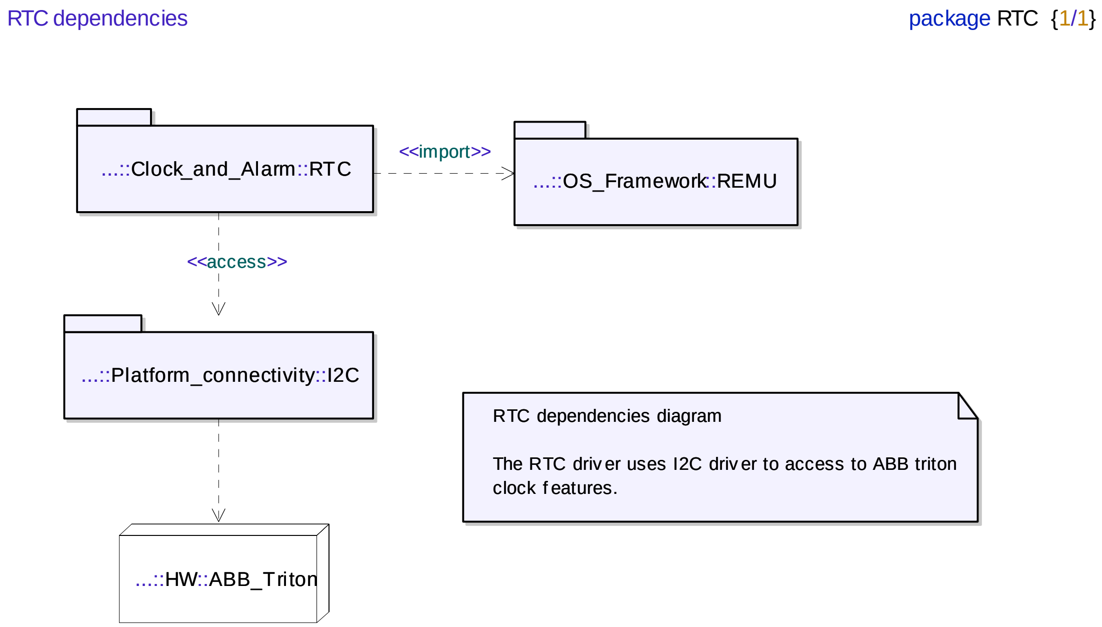

RTC dependencies
package RTC {1/1}
...::Clock_and_Alarm::RTC
<<import>>
...::OS_Framework::REMU
<<access>>
...::Platform_connectivity::I2C
...::HW::ABB_Triton
RTC dependencies diagram
The RTC driver uses I2C driver to access to ABB triton clock features.

Figure 14. RTC dependencies (Package Diagram)

3.2.1.4.2. CLMK

Purpose

The Clock Management driver low level driver

Main functions

To be completed

Restrictions

3.2.1.5. Timers

Purpose

Management of interruption, timers, watchdog.

Main Function

Restriction

3.2.1.6. DMA

Purpose

The Direct Memory Access driver is created to support the new DMA controller hardware device of the Locosto chipset in order to manage the available DMA channels. It allows to access to external RAM.

Main functions

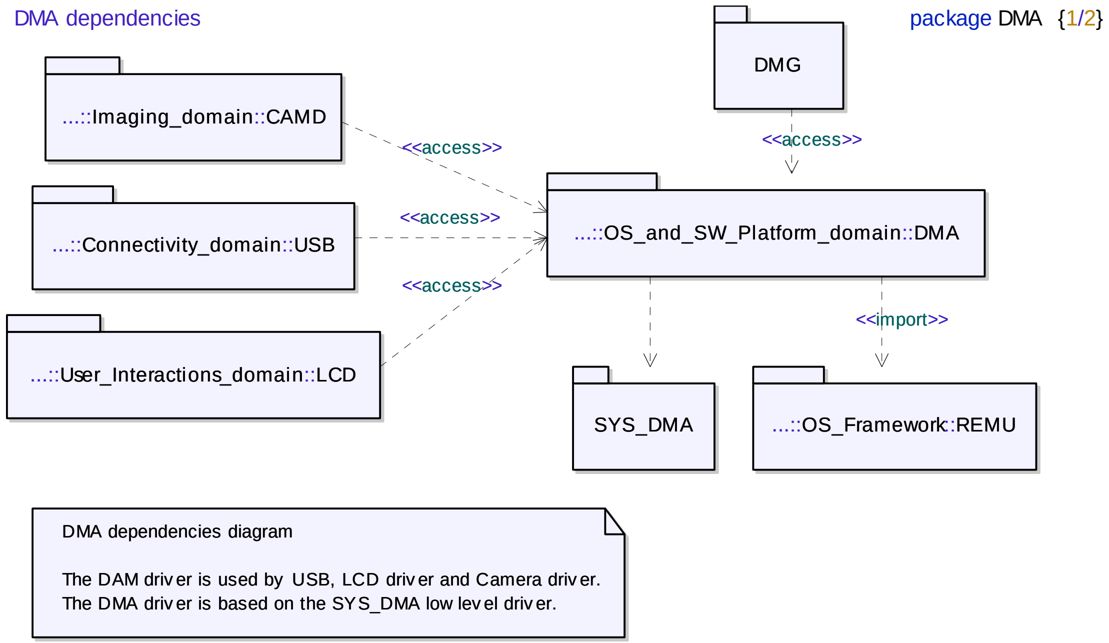
The DMA driver allows to:
- reserve a free DMA channel or a specific channel,
- remove a queued channel reservation request,
- program the specific channel parameters,
- start of a prepared DMA transfer,
- release of a reserved DMA channel

Restrictions

None

DMA dependencies diagram

The DAM driver is used by USB, LCD driver and Camera driver.
The DMA driver is based on the SYS_DMA low level driver.


DMA dependencies
package DMA {1/2}
DMG
<<access>>
...::Imaging_domain::CAMD
<<access>>
...::Connectivity_domain::USB
<<access>>
<<access>>
...::User_Interactions_domain::LCD
...::OS_and_SW_Platform_domain::DMA
<<import>>
SYS_DMA
...::OS_Framework::REMU

DMA dependencies diagram

The DAM driver is used by USB, LCD driver and Camera driver.
The DMA driver is based on the SYS_DMA low level driver.

Figure 15. DMA dependencies (Package Diagram)

3.2.1.6.1. DMG

Purpose

The DMA manager (DMG) has been created to extend the services of the DMA driver to simplify the DMA-implementation in the peripheral drivers.

The main task of the DMG SWE is to manage DMA data transfers between peripherals and internal or external memory. It allows data transfers using single and double buffering. The DMG also handles the priority arbitration of DMA-requests towards the DMA-hardware.

The DMG is a service above the DMA driver, the low-level DMA driver, and the Riviera components HISR + generic functions.

Main functions

Same as DMA, plus:
- priority arbitration : the DMG can determine the priority of the DMA request based on the peripheral ID provided by the client,
- double-buffering : the DMG takes care of switching between two buffers, not the client,
- SRAM buffer allocation : the DMG handles the copy of SRAM to External RAM,
- The possibility to transfer a large amount of data from a peripheral to XRAM in smaller chunks

Restrictions

None

DMG dependencies diagram

The DMG shall be use by applications.

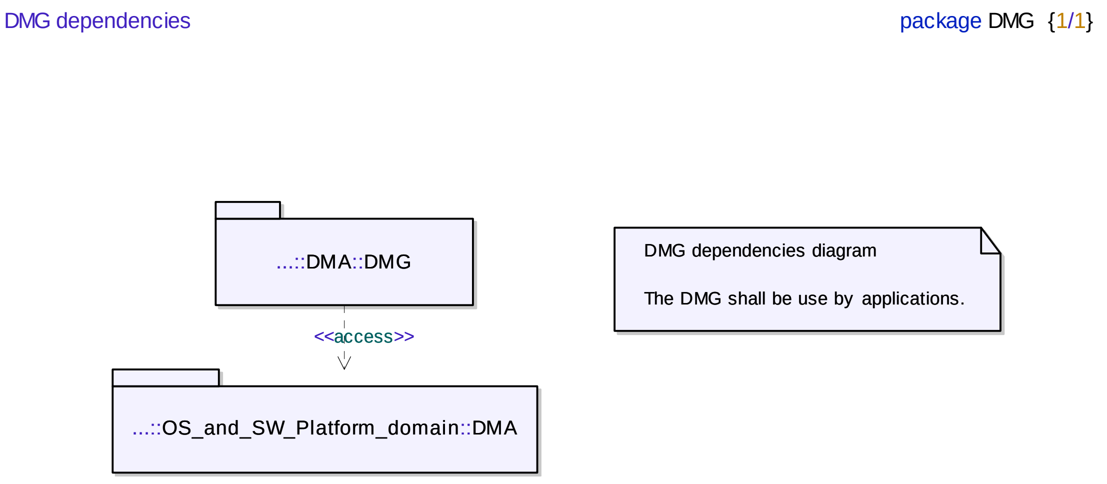


Figure 16. DMG dependencies (Package Diagram)

3.2.1.6.2. SYS_DMA

Purpose

This component provides the Low Level DMA Driver used by L1. In this case, the DMA channel is considered as hard coded DMA channel and should be used by a single component as the DMA channel for audio path.

3.2.1.7. File_System

Purpose

The File System package provides features to access to different file system with POSIX like API.
- the FFS for NOR flash
- the RS for NAND flash and MMC/SD card.

3.2.1.7.1. FFS

Purpose
The Flash File System (FFS) provides facilities for storing many kinds of data for software running on the platform. Its interface is inspired by the POSIX I/O interface.

The FFS uses the 16-Mbyte NOR Flash memory (which is also used to store the executable code) to store theses System Data.

Main functions
The list below presents all FFS functions:
- erase all data in the FFS
- format FSS

- open or create, close a file define by its pathname
- read/write data to an open file
- set file pointer of an open file
- truncate file
- flush the write buffer
- read or write file meta data
- remove object
- create, open a directory
- read an entry from a directory
- create, read a symbolic link to a file
- rename files, directories and symbolic links
- query on FSS driver
- authorize file write access

The FFS driver implements some specific global features as
- garbage collector
- dynamic object allocation
- power fail recovery

Remark: functions are asynchronous for write and synchronous for read.

Restrictions
None

FFS dependencies diagram

The FFS driver is used by:
- DAR to store recovery data
- VSI for GPF configuration
- CAMA application to store pictures.

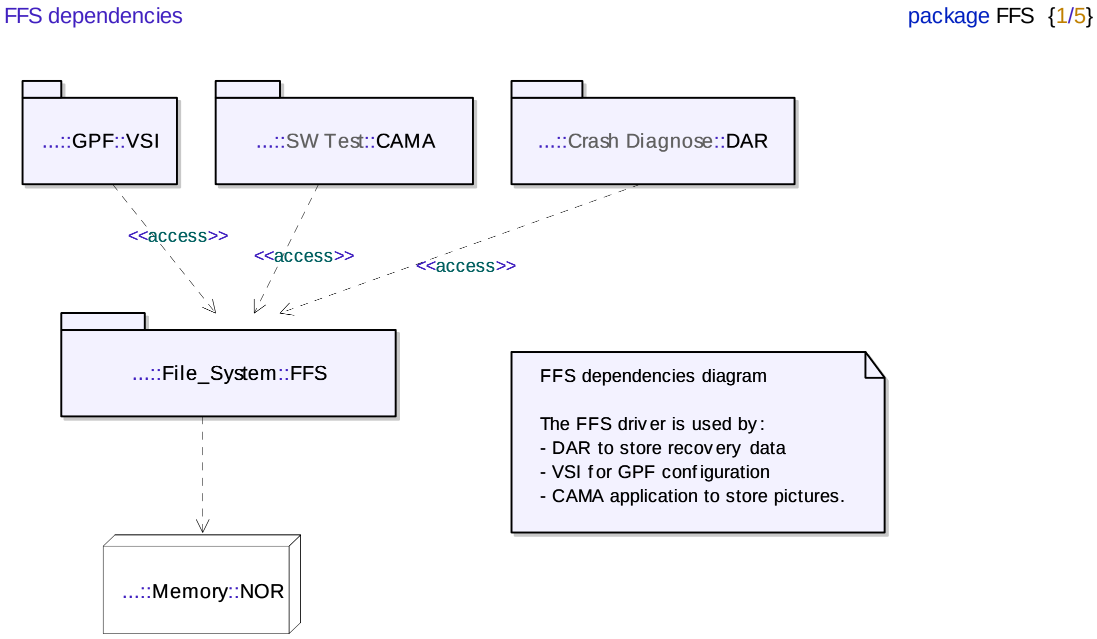


Figure 17. FFS dependencies (Package Diagram)

3.2.1.7.1.1. NOR_FFS

Purpose

The NOR_FFS driver (NOR Flash file System) is a flash file system that provides POSIX like I/O interface for a file system.

Main functions

It allows file operation such as open/read, write and delete, directory operations such as open, create and delete, symbolic link operation and file system control.

Restrictions

3.2.1.7.2. RFS

Purpose

The Riviera File System (RFS) provides unification of file access to several storage types whatever the way the blocks are organized (i.e.: file system format FAT 16, FAT 32, TI NAND FFS):
- Multi-Media Card (MMC) / Secured Digital Card (SD Card)
- Memory Stick
- NOR flash
- NAND Flash

RFS is a common file system with an API inspired by the POSIX file I/O interface. Objects in RFS are hierarchically organized in directories and sub-directories.

The application programmer doesn't have to know about the flash, in which the file is stored, because RFS is independent from the underlying flash device hardware.

RFS will mount every available device upon boot process and automatically mount removable devices on insertion and un-mount them on removal. The information whether a device is inserted or removed is provided by the GBI. When these situations occur, the RFS is responsible for passing the information to the file system cores.

Main functions

The RFS is capable to support following features:

- Well known API (POSIX look-alike)
- The File System Core that actually knows the file structure (FAT/FFS/etc..) on the media is inter-changeable.
- Support removable devices (e.g.: MMC, SD…)
- Dynamic object allocation and garbage collection
- Power fail recovery
- Wear leveling for NAND flash
- Device mounting

RFS dependencies diagram

This diagram show the internal dependencies
of the RFS driver.

RFS dependencies package RFS {1/2}

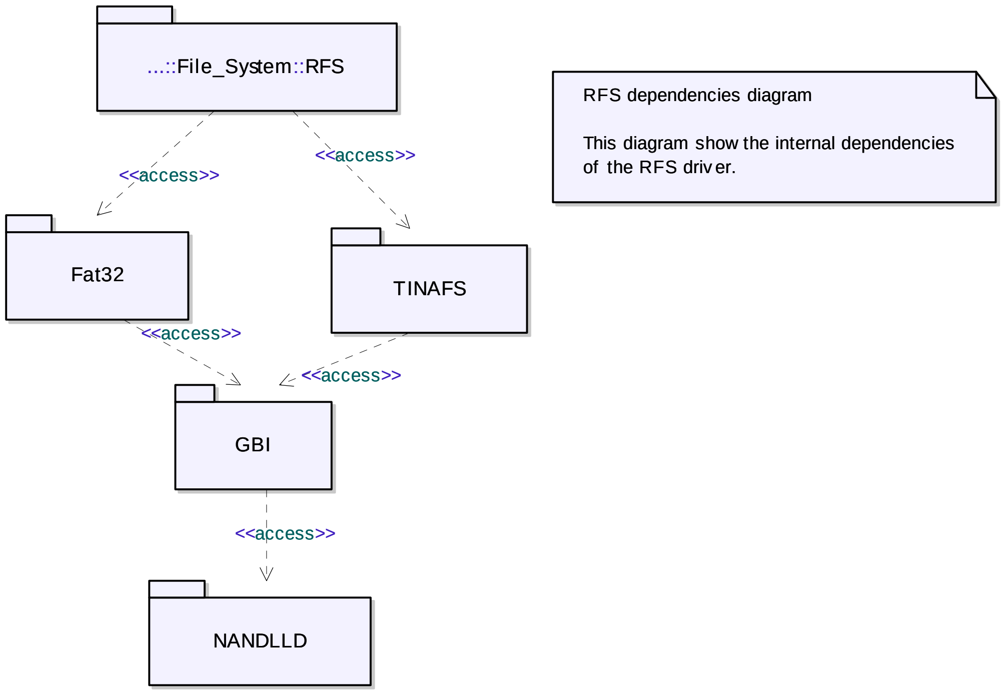


Figure 18. RFS dependencies (Package Diagram)

3.2.1.7.2.1. MMCLLD

Purpose

This SW component is not included in the Locosto delivery.

The MMC driver offers low level services for accessing MMC and Secure Digital cards which are connected to the Calypso+ MMC/SD/SDIO host controller.

Main functions
The MMC driver allows:
- to a client to subscribe to the MMC-driver in order to be able to use the driver services like reading and writing data,
- to a client to unsubscribe to the MMC-driver,
- to read and configure card registers (OCR, CID, DSR, CSD, RCA) (class 0),
- to reset cards (class 0),
- sequential or block read (class 1 and 2),
- sequential or block write (class 3 and 4),
- erasing data (class 5),
- to write protection functions (class 6),
- to set the DMA mode to be used by the driver,
- to start an identification cycle of a card stack (acquisition procedure),

- to return the number of connected MMC-cards,
- to return the relative card address of each individual MMC-card on the MMC-bus

Restrictions
Although the API supports accessing multiple MMC-cards, the driver supports a maximum of 1 MMC-card. – This is more an H/W restriction as only one controller is available –
- No support for SD specific security functions

3.2.1.7.2.2. Fat32

Purpose

The Fat32 package allows to manage the windows file system ogranization (FAT 12/16/32).

Managed Fat 12/16/32

3.2.1.7.2.3. TINAFS

Purpose

The TINAFS handles the TI NAND file system access.

Use file interface

3.2.1.7.2.4. GBI

Purpose

The requirements for a Generic Block Interface (BGI) first came into view at the moment that several devices where to be used for file storage (MMC/SD, NAND). These devices are block oriented.
From an architectural point, it is a good practice to have one software entity on top of these device drivers. This entity has a generic API and is capable of supporting various device drivers. The support for a device driver is done through a specific plugin. The different media types are that diverse, that some less generic API's are also required. They are stated as the media support subset of the GBI.
So, the GBI is an SWE with plugins to interface to various block capable device drivers.

Main functions

The GBI driver is able to:
- Do data block operations (read, write) with logical blocks. These logical blocks appear to be sequential located at the media. The real physical location of the blocks in the media is hidden and handled internally in the device driver.

- Do a data flush operation (save any unwritten data on media).
- Handle multiple media's since the Device controller can support a number of media's at the same time.
- Handle multiple partitions since each media can be divided in partitions.
- Handle media removal since some devices have removable. Subscription on insertion/removal events or initiation of media scan allows handling this.
- Perform media identification. In case of removable media, reinsertion of the same media can be recognised.
- Support quality of service. Provide information about read/write speed on the media.
- Hide any details of media specific characteristics like wearing and error correction Garbage collection, bad blocks.

For a better understanding of the GBI please note:
- The GBI organises partition information. Plugins gather information through their device drivers or static data at start-up time and (for removable media) at media insertion.
- Media and partition are uniquely identified (media ID and partition number). Clients use this identification in order to address the appropriate partition.
- Read/write data block operations are for a specific partition on a specific media.
- The GBI entity is the centre of media and partition information for client entities. Some examples of this information:
o Media type with their corresponding mountpoint names.
o File system types with their corresponding names.
Some of the information can be configured.

3.2.1.7.2.5. NANDLLD

Purpose

The NAND Flash controller driver offers straightforward access to the services offered by the hard-ware, which cover
- NAND Flash memory functions and
- Reading from the parallel port (PPI)

The NAND driver handles a 32-Mbyte NAND Flash memory, is used by the RFS driver to store user data, and provides higher speed access that the NOR memory.

Main functions

The NAND functions come from the NAND flash driver:
o read data from NAND Flash and optionally computes ECC values,
o write a NAND Flash page and optionally computes ECC values,
o copy a memory area to another area,
o erase a given memory block,
o read the NAND Flash status register,
o read the NAND Flash id,

o reset the NAND Flash,
o check the validity of a block according to the manufacturer

- PPI functions
o read data from the parallel port,
o read continuously data from the parallel port,
o select the DMA mode to be used by the driver,
o return the state of the NAND Flash controller

Restrictions

- NAND functions
o There are no sequential read modes, since current hardware does not support them.
o Only SAMSUNG K9F5608xx devices are supported.

- PPI functions
o Since the driver will only handle one request at a time, NAND Flash requests cannot be handled during a parallel port transfer.

3.2.1.7.2.6. FS-Core

Purpose

The FS-core defines generic FS-core types.

The intended user of the FS-core specific API's will be the RFS. This RFS interface provides a unified access to media devices. It gathers the applications requirements for a common File System.

The RFS can also inform the appropriate FS-core of changes, by sending the core a new partition/mount point table.

The two FS-cores used are FAT and NAND.

Main functions

The FS-core driver allows to:
- Inform the according FS-core of a new partition table (list of partition information),
- return a pointer to the FS-core function table

3.2.1.8. Boot

3.2.1.8.1. RHEA

Purpose

The RHEA driver allows to configure the number of wait state and time-out on the internal bus.

Mains function

NEED MORE INFORMATION
Add management of RHEA switch:
- configuration of RHEA switch
- dynamic management (MIDI vs MP3 on C-port).(ARM7/DSP).
3 functions -> porting for new configuration.

Restriction

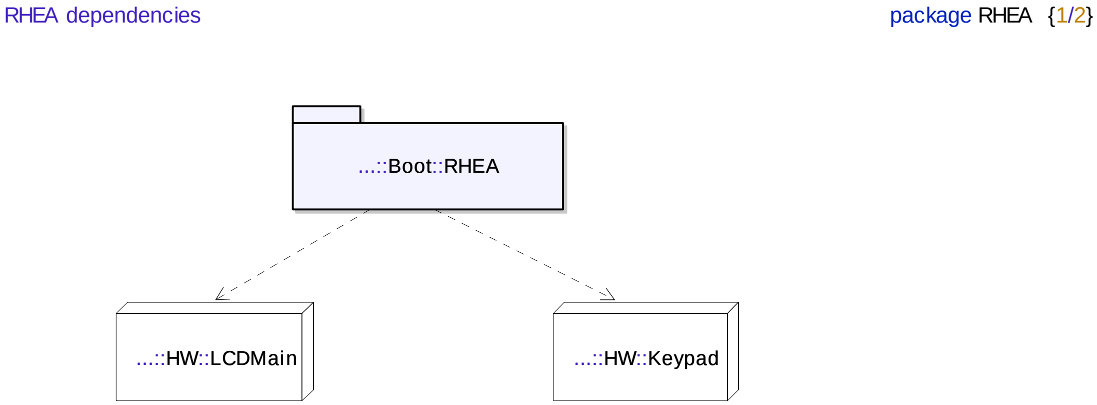


Figure 19. RHEA dependencies (Package Diagram)

3.2.1.8.2. INTH

Purpose:

This Interrupt driver (INTH provides an abstraction between the hardware registers and the application layers. It allows to manages to type of interruption: IRQ and FIQ (Fast Interruption reQuest).

There is 32interruption allowed on the manin level and 5 on the second level.

The interruption supports the level or edge sense.

SW Components : drv-core/inth/sys_inth.*

INTH dependencies diagram

The INTH driver is used by

- LCD driver
- DMA driver

To be competed.

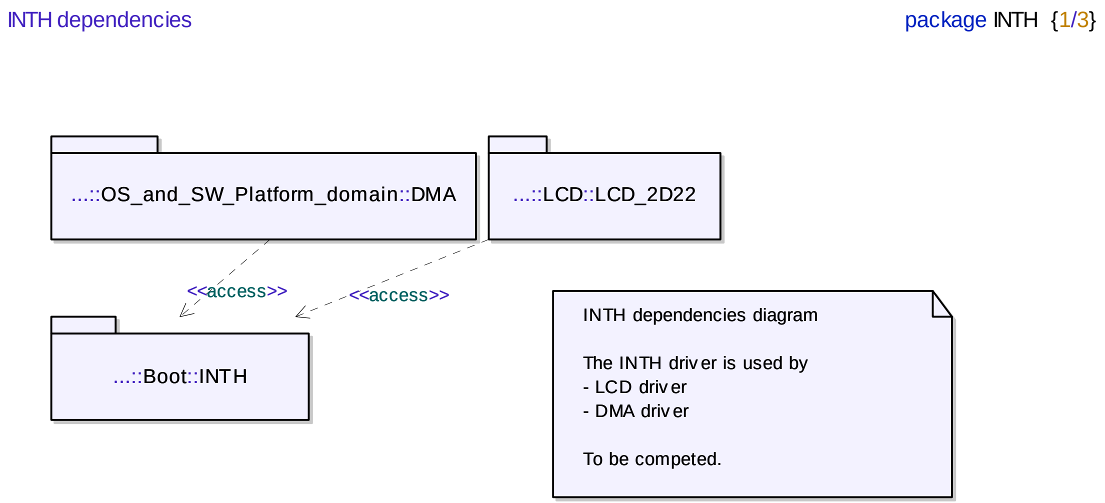


Figure 20. INTH dependencies (Package Diagram)

3.2.1.8.3. MEMIF

Purpose

The Memory Interface driver connects the processor to internal/external memory.

Main functions

Restrictions

3.2.1.9. Inter_domain_communication

3.2.1.9.1. CCI

Purpose

Provide function to send samples to the C-Port of the ABB

Main functions

To be completed

Restrictions

3.2.1.9.2. SOFTCSMI

Purpose

The multi-channel shared memory driver (CSMI) provides agnostic interface between Application domain and modem domain.

Main Functions

The SCSIMI driver allow receiving amd sending data over several streams.
IT should run on differents OS, that support thread.

Restrictions
This driver is not used in Locosto anchor 1. BMI/MFW are connected diectly to ACI.

Pending question: Is it a mandatory driver for Locosto?
For GPRS reference anchor only.

To check: name convention: SoftCSMI or IDC.
SoftCSMI dependencies diagram

The SoftCSMI is the SW implementation of the CSMI driver used by Neptune in order to keep consistency between the two platforms.

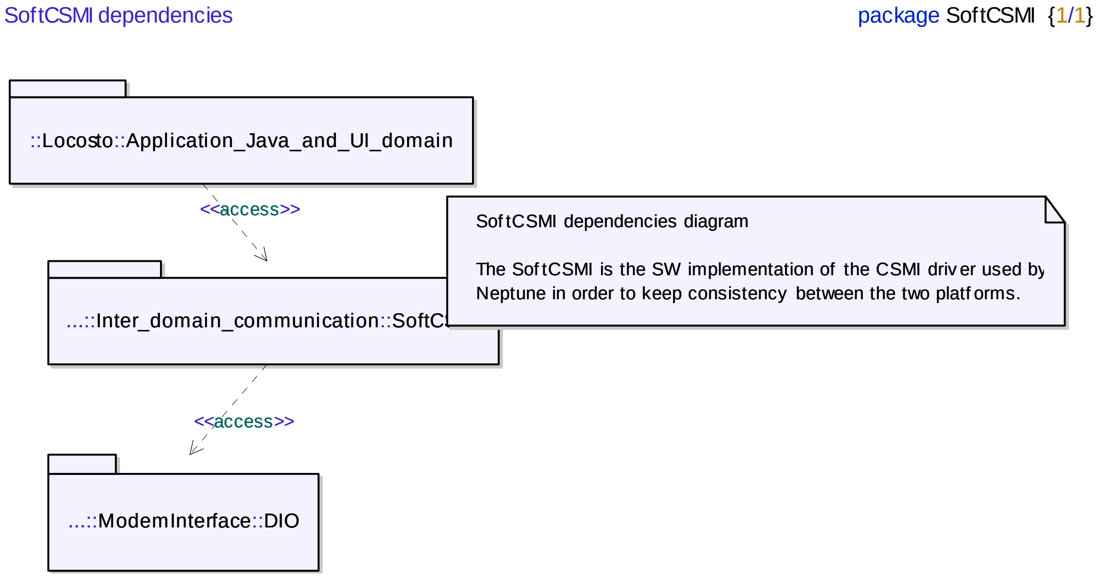

SoftCSMI dependencies
package SoftCSMI {1/1}
::Locosto::Application_Java_and_UI_domain
<<access>>
...::Inter_domain_communication::SoftC
SoftCSMI dependencies diagram
The SoftCSMI is the SW implementation of the CSMI driver used by Neptune in order to keep consistency between the two platforms.
<<access>>
...::ModemInterface::DIO

Figure 21. SoftCSMI dependencies (Package Diagram)

3.2.1.10. Platform_connectivity

3.2.1.10.1. SPI

Purpose

The Serial Port Interface is a bi-directional interface dedicated to the transfer of data from up to five external devices offering the same interface.

Main Functions

The SPI provides an interface to access the ABB hardware components like Power ON/OFF management, Analog to Digital Conversions, SIM control, Battery Charging control and LEDs control. Since several entities in the TI GSM/GPRS SW can access to the ABB at any moment, it is important to prevent concurrent access.

To protect ABB access, two different mechanisms have been introduced. The first one is an SPI task that allows scheduling accesses sequentially. It is highly recommended to use this SPI task through its API. The SPI task priority is one of the most important in the whole system. The second one is a semaphore protection for direct accesses to the low-level driver.

Consequently, the architecture of the SPI driver consists in three components:
- the SPI Driver API, called by any SW entity and sending messages to the driver task
- the SPI task and its mailbox, receiving messages from the Driver API and calling the low-level driver functions
- the low level driver, which is a set of functions performing access to the hardware SPI module to read or write data in the ABB registers

Restrictions

In TI Chipset solution, the SPI is exclusively used to connect the TI Analog Base Band (ABB). There-fore, it is assumed that the ABB is connected as the SPI device 0.

Pending Question: To check on SOC: perhaps rework for simplification

3.2.1.10.2. GPIO

Purpose

The General Purpose Input/Output driver managed
- external peripheral in output
- interrupt in input (jog dial)

Support 3 instances:
-

Main function

Restriction

SW component: drv-core/armio

3.2.1.10.3. I2C

Purpose

The I2C driver is in charge of the configuration of the ABB Triton. It provides high level services for communication with I2C devices which are connected to the I2C controller of the LoCosto chipset.

It allows to control the on the ABB of:
- Power,
- USB transceiver
- Voice and stereo codec's

The driver supports Multi-master transmitter/receiver communication mode but does not support Slave receiver/transmitter communication mode. This means that the LoCosto will always be master.

The driver supports 7-bit and 10-bit addressing mode which is configurable at compile time. The driver support standard mode (up to 100 Kbits/s) and fast mode (up to 400 Kbits/s) which is configurable at compile time.

Main functions

The I2C driver allows:
- To select the Transfer mode to be used by the driver (interrupt or polling),
- To read a number of bytes from an i2c address,
- To write a number of bytes to an i2c address

Restrictions

Pending question: Multiclient, priority management. Not managed in C+

I2C dependencies diagram

The I2C driver is used to communicate with ABB Triton
registers.

I2C dependencies package I2C {1/2}

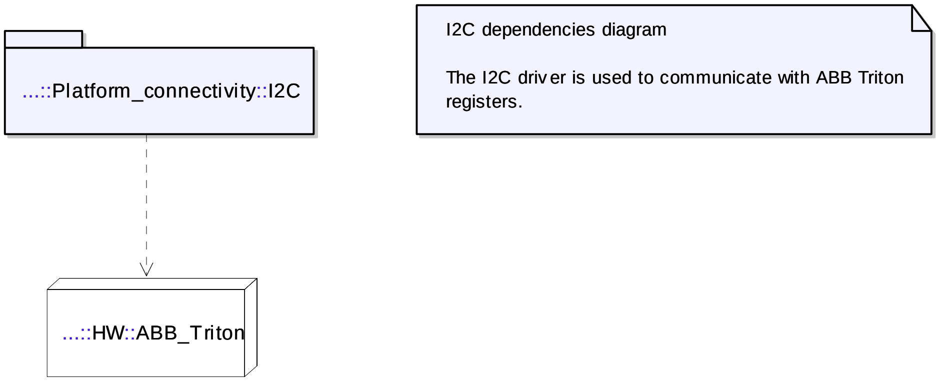

...::Platform_connectivity::I2C
I2C dependencies diagram
The I2C driver is used to communicate with ABB Triton registers.
...::HW::ABB_Triton

Figure 22. I2C dependencies (Package Diagram)

3.2.2. Power_Management_domain

Purpose

See DAD power management

3.2.2.1. Battery Power Mgt

3.2.2.1.1. LCC

Purpose

The Low Cost Charger module (LCC) is a generic driver for battery and charger devices. Based on RVF.

Main functions

The LLC module supports the following functionality:
- battery supervision: temperature, capacity and voltage
- battery charging of Lithium-ion (LiIon) or Nickel Metal-Hydrur (NiMH) based battery types
- battery charging using constant current (CI) charger
- battery charging using constant voltage (CV) charger
- battery charging using unregulated (low cost) charger

TI DK: Brian Jorgensen

3.2.2.2. Power Management Fwk

3.2.2.2.1. ULPD

Purpose

The Ultra Low Level Power down driver allows to control the ULPD controler. It is the interface of the Power Management and Clock management running on the DBB Triton.

Main functions

The main functions of the ULPD block are:
- Gauging of the 32 kHz Clocking signal supplied from ABB Chip.
· Maintenance of GSM time during deep-sleep mode with the minimum time accuracy to allow a burst demodulation at wake-up
· Programmable timer to exit deep-sleep
· Delivery of the 13MHz master clock to the CLKM/DPLL modules
· Switching between 13MHz and 32KHz

Restrictions

SW Component

drv-core/ulpd
ULPD dependencies diagram

To be completed

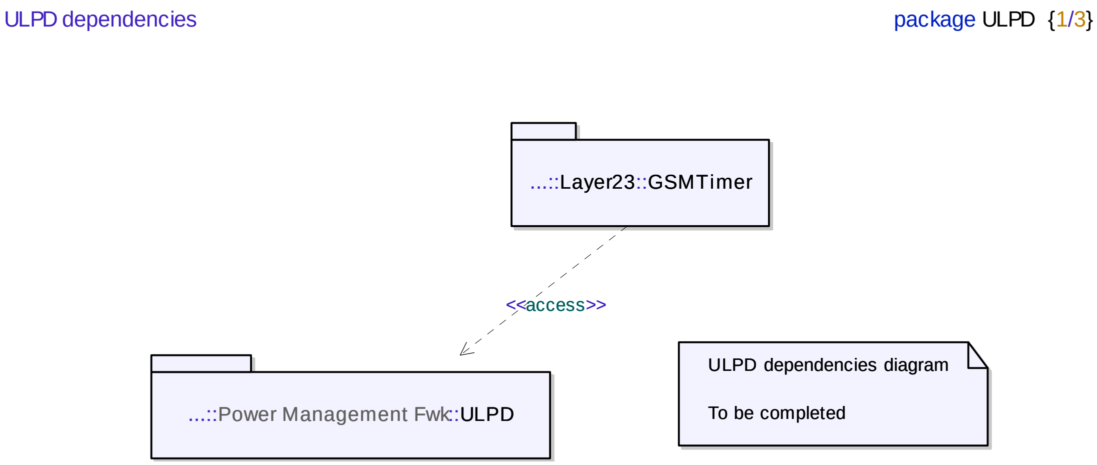


Figure 23. ULPD dependencies (Package Diagram)

3.2.2.2.2. PWR

Purpose:

The Power driver (PWR) provides the main functions to control the main power of the DBB.

Main functions:

Restrictions:

SW Component: pwr_

To check: Management on ON/OFF

3.2.3. Modem_domain

The Modem package contains all SW packages involved in GSM/GPRS features.

It contains:
- the Protocol Stack Interface (PSI)
- the AT Command Interface (ACI)
- the Layer 23 (L23)
- the Layer 1 (L1)

All this software packages are build on GPF frame.

3.2.3.1. ModemInterface

Purpose

This packages give a view of all software components involved in the Modem Interface. These components don't stand in the ACI package.

Modem Interface dependencies diagram

This diagram shows that DIO and PSI have connection interface with the modem domain and the BSP domain.

DIO is only an interface for drivers.

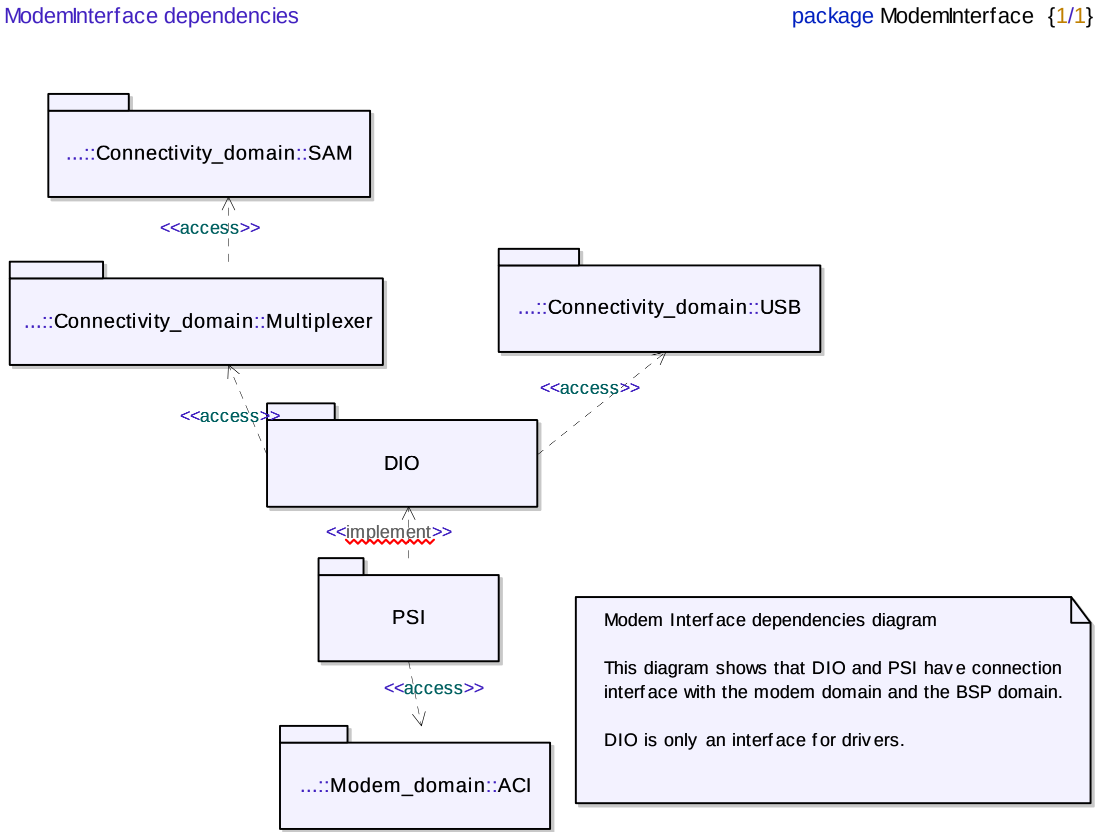

ModemInterface dependencies
package ModemInterface {1/1}
...::Connectivity_domain::SAM
<<access>>
...::Connectivity_domain::Multiplexer
...::Connectivity_domain::USB
<<access>>
<<access>>
DIO
<<implement>>
PSI
<<access>>
...::Modem_domain::ACI
Modem Interface dependencies diagram
This diagram shows that DIO and PSI have connection interface with the modem domain and the BSP domain.
DIO is only an interface for drivers.

Figure 24. ModemInterface dependencies (Package Diagram)

3.2.3.1.1. DIO

Purpose

The Data Inpout/Output package provides functional interface to the device drivers.

Main functions

It allows access to different devices drivers with the same API. It provides functions to
- initialization and function call routing. That means initialize drivers and establish a link between the DIO drivers and the protocol stack.
- callback routine
- user/interface termination

Each driver that want to access to the protocol and reverse should implement the DIO interface.

Restrictions

3.2.3.1.1.1. DIO_IL

Purpose

The DIO Interface Layer (DIO_IL) is the API that should implement a devices driver that wants to access/call to/by the protocol stack.

3.2.3.1.2. PSI

Purpose

The Protocol Stack Interface (PSI) provides a single API to access to the protocol stack features.

Main Functions

The exchanges between PSI and ACI are message based.

Restrictions

3.2.3.2. ACI

Purpose

The Application Control Interface (ACI) provides the functionalities of the protocl stack to the applications via an AT command interface.

Main Function

The protocol stack (L23) services: (see the list of SW in PSA)
- The call control: incoming/outgoing call, emergency call
- The packet services: activating/deactivating Packet Data Protocol (PDP) context.

The Data Connection manager (DTI)

Restrictions

SW Component
PSA dependencies diagram

This diagram shows the internal dependencies of the ACI package:
- PSA deals with L23
- Application uses BAT or ATI to access to PS features (as MFW)
- DTI is used by L23 to access to Device driver.

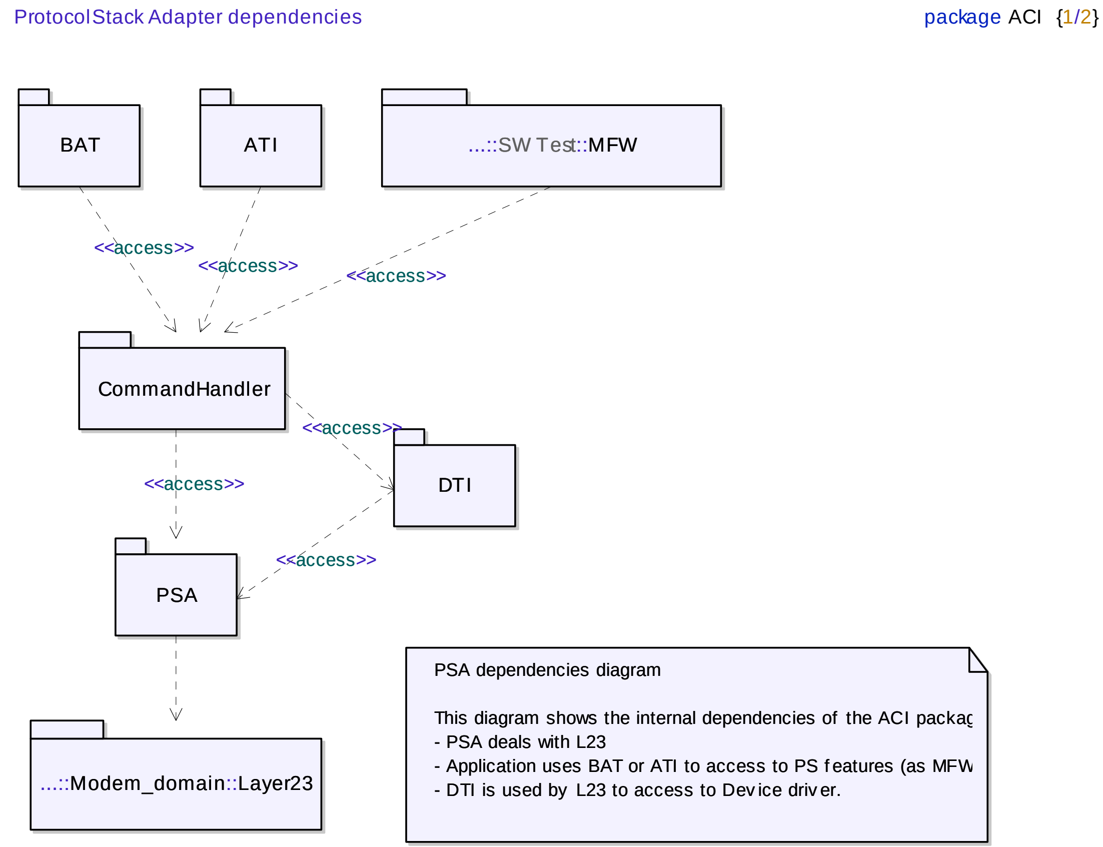

ProtocolStack Adapter dependencies
package ACI {1/2}
BAT
ATI
....::SW Test::MFW
<<access>>
<<access>>
<<access>>
CommandHandler
<<access>>
DTI
<<access>>
<<access>>
PSA
...::Modem_domain::Layer23
PSA dependencies diagram
This diagram shows the internal dependencies of the ACI packag
- PSA deals with L23
- Application uses BAT or ATI to access to PS features (as MFW
- DTI is used by L23 to access to Device driver.

Figure 25. ProtocolStack Adapter dependencies (Package Diagram)

3.2.3.2.1. ATI

Purpose

The AT-Interpreter (ATI) provides function to parse AT command from several parallel channels of AT command stream.

Main function

The commands are checked against the AT command syntax and parameters are converted from string to binary representation. Also line edit functionality is provided for editing commands, or complete SMS messages from a PC via the serial/USB interface.

Restrictions

3.2.3.2.2. BAT

Purpose

The Binary AT command interface (BAT) provides a binary interface to send AT command from the Application domain to ACI.

Main functions

To be completed

Restrictions

3.2.3.2.3. COMMANDHANDLER

Purpose

The Command Handler (CM) handles the main processing of the ACI and contains the state machine that deals one or several Protocol Stack Adapter.

Main functions

Restrictions

3.2.3.2.4. DTI

Purpose

The DTI-Manager (DTI) is in charge of establishing supervising and deactivation of the right dynamic links between entities in order to build data stacks

Main function

It uses an message interface, it allows to exchange buffers.

To be completed

Restriction

3.2.3.2.5. PSA

Purpose

The Protocol Stack Adapter (PSA) is in charge of exchanges with the L23 primitives.

Main functions

To be completed

Restrictions

SW Component:

3.2.3.2.5.1. SMS

See Layer23 SW component.

3.2.3.2.5.2. SS

See Layer23 SW component.

3.2.3.2.5.3. CC

See Layer23 SW component.

3.2.3.2.5.4. L2R

See Layer23 SW component.

3.2.3.2.5.5. MM_GMM

See Layer23 SW component.

3.2.3.2.5.6. T30

See Layer23 SW component.

3.2.3.2.5.7. PPP

The Point to Point Protocol allows to managed transfer of level 2. The client mode
See Layer23 SW component.

3.2.3.2.5.8. SIM

See Layer23 SW component.

3.2.3.2.5.9. SM

3.2.3.2.5.10. PHB

See Layer23 SW component.

3.2.3.2.5.11. RR

See Layer23 SW component.

3.2.3.2.5.12. EM

See Layer23 SW component.

3.2.3.2.5.13. TCSD(?)

Transparent Circuit Switched Data manages the primitive related to TCSD data.

See Layer23 SW component.

3.2.3.2.5.14.   SNDCP

See Layer23 SW component.

3.2.3.2.5.15.   DTI

See Layer23 SW component.

3.2.3.3.   Layer23

Purpose

The Layer 23 manages several entities to handle GSM/GPRS features:
- the Circuit switch data
- the GSM core
- the GPRS core

Layer23 dependencies diagram

To be completed

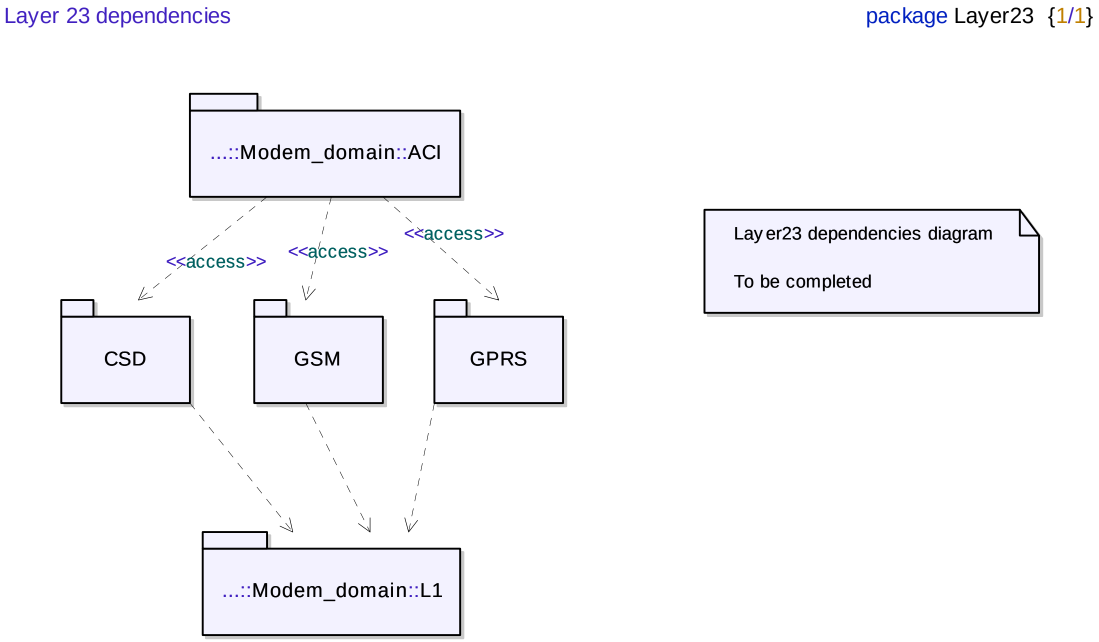


Figure 26. Layer 23 dependencies (Package Diagram)

3.2.3.3.1.   CSD

Purpose

The Circuit Switch Data (CSD) package is a logical package that contains all SW components involved in the CSD features. It extends the GSM core with the circuit switched data services transparent, non-transparent and FAX-transfer/reception with a speed up to 14.400 bps.

CSD dependencies diagram

To be completed

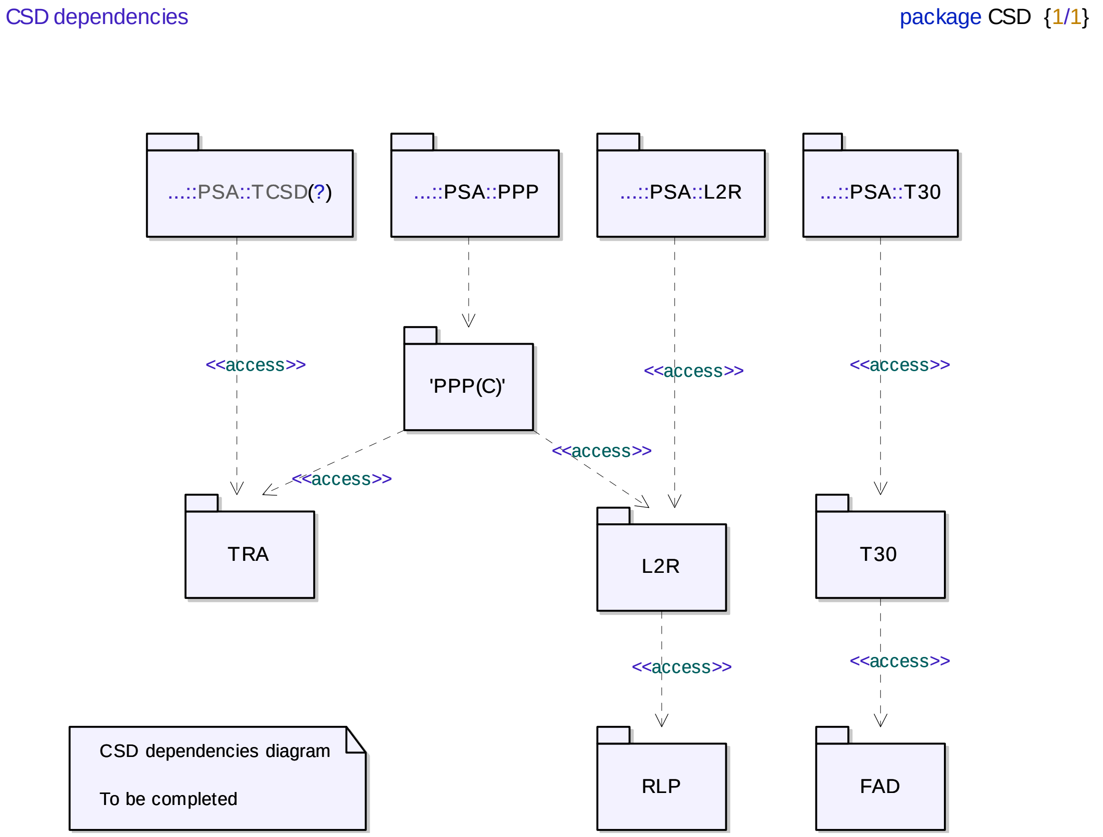


Figure 27. CSD dependencies (Package Diagram)

3.2.3.3.1.1. PPP(C)

Purpose

The Point to Point Protocol (PPP) can run in a server and in a client mode. PPP allows to perform a authentication between server and client using CHAP or PAP authentication protocols. Once the connection is established PPP allows packet oriented communication via a serial connection. In case of circuit switched data communication PPP is used on top of L2R/TRA to authenticate the MS in the network. In GPRS it acts as a server, providing a circuit switched like interface to a connected PC running dial-up network.

3.2.3.3.1.2. L2R

Purpose

The Layer 2 Relay (L2R) converts between the packet oriented RLP service and the serial data coming from PPP or from a serial link.

3.2.3.3.1.3. FAD

Purpose

The FAX Adapter (FAD) is required to convert analogue based signaling and data flows in a fax connection to the digital radio interface since FAX is not used over a voice channel in GSM.

3.2.3.3.1.4. T30

Purpose

This entity implemented the T.30 standard for FAX transmission. It allows connecting to a remote FAX machine or a FAX modem and to send/receive and poll T.4 coded pages. Together with the T.32 compliant FAX AT-command set that is implemented in ACI the MS can work as a FAX class 2.0 modem connected to a PC or to a internal FAX application.

3.2.3.3.1.5. TRA

Purpose

The Transparent data (TRA) service allows to use a unprotected CSD connection to the network. It converts the serial data from UART or PPP to the block oriented CSD interface of L1.

3.2.3.3.1.6. RLP

Purpose

The Radio Link Protocol (RLP) is the layer 2 of the non-transparent circuit switched data. The entity provides a reliable data connection service.

3.2.3.3.2. GSM

Purpose

GSM core: GSM MO/MT voice calls + MO/MT SMS services including MT Cell broadcast decoding.
GSM dependenecies diagram

To be completed

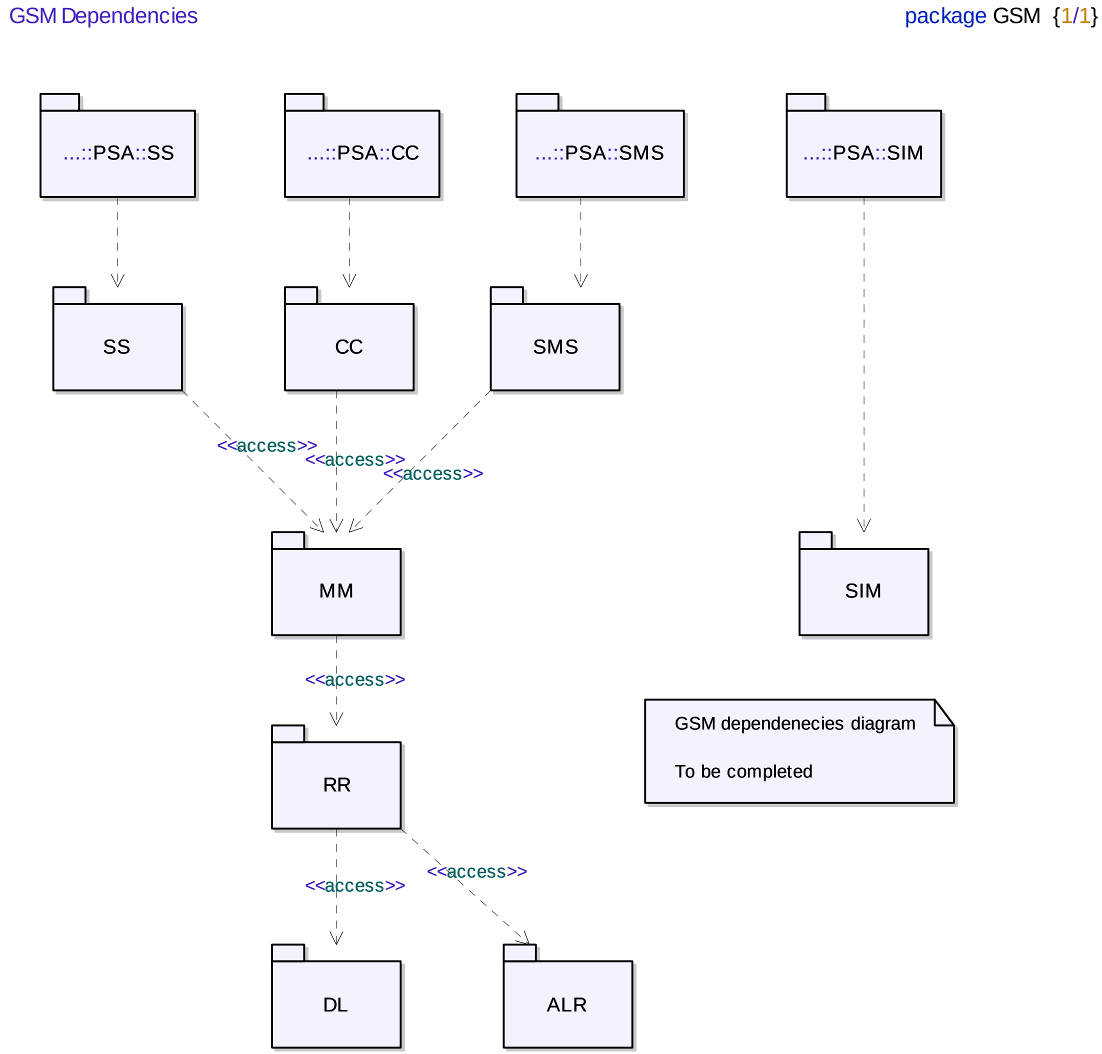


Figure 28. GSM Dependencies (Package Diagram)

3.2.3.3.2.1. ALR

Purpose:

The Adapter Layer (ALR) allows communication between RR and L1. It supports RR in cell-selection/re-selection, SI preprocessing and paging detection.

Main functions:

To be completed.

Restrictions:

To be completed:

SW Component:
- source : alr
- library: alr.lib

3.2.3.3.2.2. DL

Purpose:

The Data Link Layer which provides layer 2 functionality to RR on different logical GSM channels.

Main functions:

To be completed.

Restrictions:

To be completed.

SW Component:

3.2.3.3.2.3. SIM

Purpose

The Subscriber Identity Module (SIM) controls the SIM driver manages the access to different SIM data fields and provides a SAP towards MM, GMM and ACI.

3.2.3.3.2.4. RR

Purpose

The GSM Radio Resource (RR) takes care that a suitable cell is selected, that the surrounding neighbor cells are observed and that the serving cell with the best radio quality is used, in idle mode and in a call or data connection.

3.2.3.3.2.5. CC

Purpose

The Call Control (CC) is in charge of circuit switched call handling. It performs the required signaling between MS and network in order to establish, receive, maintain and end a call. Handling of conferences and 2nd call is also in the scope of this entity. CC is part of the connection management layer.

3.2.3.3.2.6. SS

Purpose

The Supplementary Service (SS) is in charge of getting/setting and querying services in the network like call-forwarding, call-deflection and call-barring. SS is part of the connection management layer.

3.2.3.3.2.7. SMS

Purpose

The Short Message Service (SMS) provides the capability to send and receive SMS. The reception of Cell Broadcast Messages (CBM) is also handled here. SMS is part of the connection management layer.

3.2.3.3.2.8. MM

Purpose

The Mobility Management (MM) is in charge to take care that the mobile stays in the registered state on the home or roaming network. It initiate the cell selection and informs the network if a cell reselection or a hand-over changed into a different location area. MM maintains the full or limited service of the phone.

3.2.3.3.2.9. LC

Purpose

These entities are the AGPS subsystem which is described in [6]. Controlling the TI GPS chipset via a I2C connection and using the assistance data sent by the network during a location procedure the subsystem can provide position fixes which are sent over the network or towards the application. The subsystem supports MS based, MS assisted and standalone position fixes.

1.1.1.1.1.1.1 GPSM

3.2.3.3.2.10. RRLP

See LC.

3.2.3.3.2.11. GPSM

See LC.

3.2.3.3.3. GPRS

Purpose

The GPRS package is a logical package that contains all SW components involved in the GPRS functionalities. It extends the GSM core with packet switched data service GPRS up to class 12 which means for transfer 5 timeslots in the TDMA raster can be used in parallel (4RX max / 4TX max but number of slots should not exceed 5).

GPRS dependencies diagram

To be completed

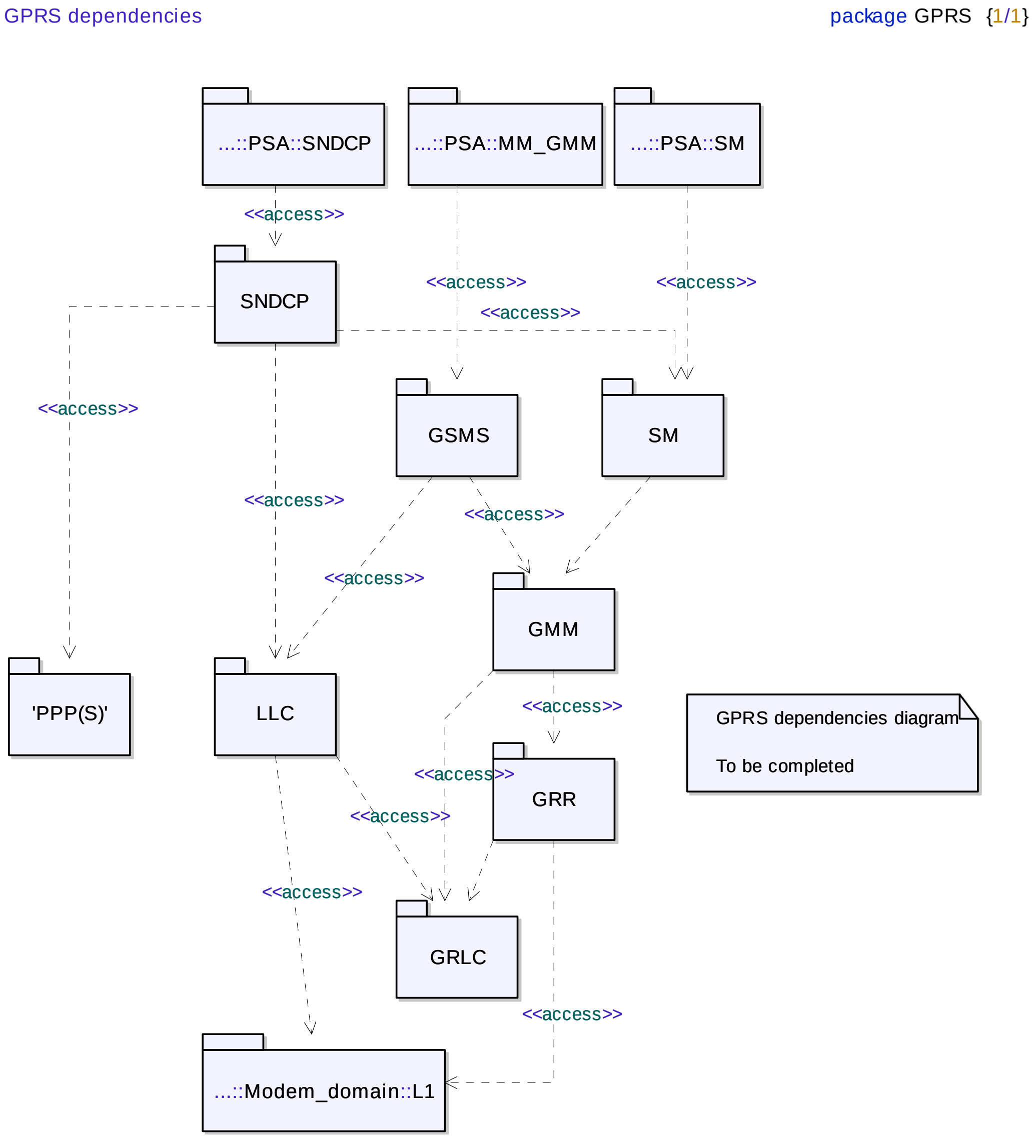


Figure 29. GPRS dependencies (Package Diagram)

3.2.3.3.3.1. GSMS

Purpose

With GPRS it is also possible to use a packet oriented connection to send and receive SMS. GSMS is not a entity on its own. It should be considered more as the functional extension of the SMS entity towards GPRS. GSMS is part of the connection management layer.

3.2.3.3.3.2. SNDCP

Purpose

The Subnetwork Dependent Convergence Protocol (SNDCP) entity provides the packet oriented access point for e.g. TCP/IP and PPP services (NSAPI). Dependent on the QoS of each PDP connection SNDCP performs the prioritization and segmentation/reassembly of LLC frames. Furthermore SNDCP performs IP header compression and V.42bis data compression.

3.2.3.3.3.3. SM

Purpose

The Session Management (SM) controls the activation/deactivation and modification of the PDP (Packet Data Profile) context in the GPRS system. A context defines the QoS for a packet oriented connection, the used NSAPI/SAPI, the IP addresses of the MS, Gateway and the DNS. Up to 7 context can be handled by SM. SM is part of the connection management layer.

3.2.3.3.3.4. GMM

Purpose

The GPRS part of mobility management extend the MM functionality in that way that it takes care that the suitable cell is chosen in order to fulfill the required GPRS operation. A change of a routing area is notified to the network by this entity.

3.2.3.3.3.5. LLC

Purpose

The Logical Link Control (LLC) establishes and maintains the logical links (SAPI) between the GPRS network and the MS. It can operate in acknowledged and unacknowledged mode and provides several logical channels with different priorities to allow traffic with different QoS for control and data messages. LLC uses the GEA hardware block in order to cipher/decipher the data packets and to speedup the FCS calculation of these blocks.

3.2.3.3.3.6. GRR

Purpose

The GPRS Radio Resource takes care that on the new GPRS channels PBCCH/PCCCH a suitable cell is selected, that the surrounding neighbor cells are observed and that the serving cell with the best radio quality is used, in idle mode and in a call or data connection. Furthermore GRR establish and terminates the GPRS packet transfers called Temporary Block Flow (TBF) which is performed by GRLC together with the MAC in L1.

3.2.3.3.3.7. GRLC

Purpose

The GPRS Radio Link Control (GRLC) entity performs the TBF of the packet oriented GPRS traffic. It can operate in acknowledged and unacknowledged operation mode. GRLC maintains a send and a receive window. Via the radio priority GRLC blocks can be send in a certain order. GRLC contains also the upper MAC of the GPRS stack.

3.2.3.3.3.8. PPP(S)

Purpose

To be completed

3.2.3.3.4. CST

Purpose: TO BE COMPLETED
AEC_Enable
Addition AT commands to:
- Get ADC values
- Control AEC and noise reduction parameters
- Perform soft reset of the ARM7

Main functions:

Restrictions:

Software component:
cst

3.2.3.3.5. GSMTIMER

Purpose

To be completed

3.2.3.3.6. L2DATA

Purpose

To be completed

3.2.3.4. L1

Purpose

The L1/MCU part acts as a master to ontrol all signal processing task that are provided by the L1/DSP part.
L1 dependencies

To be completed

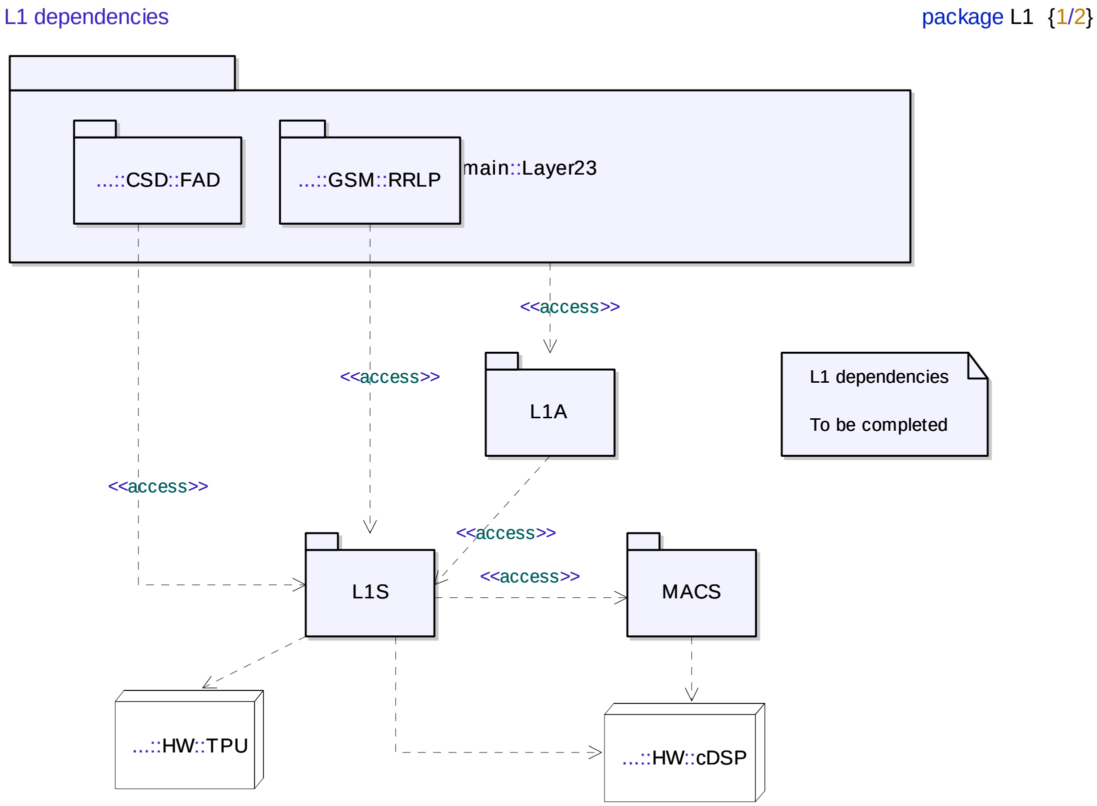


Figure 30. L1 dependencies (Package Diagram)

3.2.3.4.1. MACS

Purpose

Medium Access Control Synchronous (MACS) is specialized to handle the transmission of uplink/downlink RLC/MAC blocks and controls the DSP tasks which are responsible signal processing (coding/decoding) during GPRS packet transfer mode. MACS interfaces with the upper layer entity via dedicated function calls, which are implemented by GRLC. This interface is solely dedicate to exchange RLC/MAC control and data blocks. All other communication with the network side will be handled by L1S. MACS will be called by L1S on each TDMA frame during packet transfer mode to obtain allocation information which will be used for TPU scenario programming. To access and control the signal processing tasks, MACS uses the DSP API shared memory interface.

3.2.3.4.2. L1A

Purpose

Layer 1 Asynchronous (L1A) is the asynchronous part of the of the L1/MCU software. It owns the message based primitive interfaces MPHC and MPHP for the communication with the Layer 2 and 3 part of the GSM/GPRS modem software. L1A forwards the service requests from the upper layer entities to the L1 synchronous part to request the scheduling of those services by the DSP software. The communication with the synchronous part of L1 is achieved through a shared memory interface, which has to be set up with the right information prior to the next scheduling of the L1 synchronous task itself. In contrast the return path is implemented by a message based system queue, which will hold the results and acknowledgements that will be placed by L1 synchronous part. L1A consists of several dedicated state machines for certain modes of operations (e.g. cell selection, idle mode and dedicated mode).

3.2.3.4.3. L1S

Purpose

Layer 1 Synchronous (L1S) is the synchronous part of the L1/MCU software and will be scheduled automatically on the beginning of each TDMA frame. The L1S is implemented a High Level Interrupt Service Routine (HISR) and will preempt all other activities that are currently running. The L1 synchronous part has two main tasks to perform. One of them is to control and schedule the signal processing tasks of the DSP to perform the requested L1 services. These are for example, measurement requests, channel reading, serving cell or neighbor cell synchronization. To access and schedule the signal processing tasks, L1S uses the DSP API shared memory interface. Every service and therefore every task has its own priority, which has a direct influence on the scheduling algorithm. The L1S uses a functional interface to directly communicate with upper layers for the exchange of GSM control messages and uplink/downlink data buffers for circuit switched data. All other communication with the upper layers is done through the L1A system queue. The other main task of L1S is the programming of scenarios for the Time Processing Unit (TPU). The TPU is a hardware coprocessor for the GSM time base with a quarter bit precision. It will be used to synchronize and time align the uplink bursts with the GSM/GPRS system air interface. For communication with the TPU a driver is used in conjunction with a shared memory area to hold the TPU scenarios.

3.2.3.4.4. DSPDWMLD

Purpose

The DSP download (DSP_DNWLD) allows to load patch on the DSP.

In this module, API stands for Access Port Interface, the shared memory between the ARM processor and the DSP.

3.2.3.5. CCD

Purpose

Air interface message coding decoding.

3.2.3.6. CC

3.2.4. Security_domain

Security dependencies diagram

RSS/RSS Services provided secure service for:
- secure boot rom
- security functions

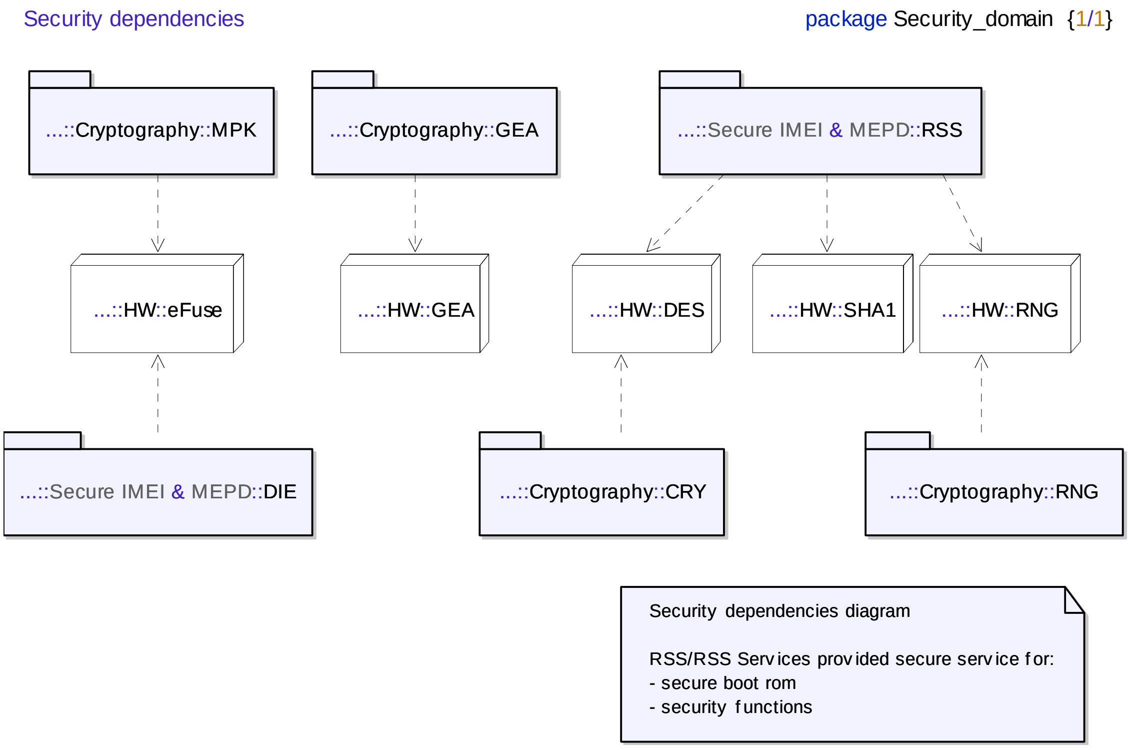


Figure 31. Security dependencies (Package Diagram)

3.2.4.1. Secure IMEI & MEPD

3.2.4.1.1. DIE

Purpose

The DIE driver (DIE) provides a service to read the 64 bit DIE ID register of the Calypso+ chipset. The Die ID cell is a 64-bit register composed of fuse cells electrically programmed during the manufacturing process. Originally intended for engineering purpose, each Die ID being unique can be used for application requiring the authentication of the device.

Main functions

The DIE driver allows to:
- return the DIE id 64-bit value

Checks for porting. 128 bits

Figure 32. DIE dependencies (Package Diagram)

3.2.4.1.2. RSS

Purpose

The driver offers services for using the security related services coded into the ROM of the Calypso+ chip.

These services are:
- Run-Time Loader (binding)
- Run-Time Loader (unbinding)
- Run-Time Checker
- Run-Time Platform Data Checker

Main functions

The RSS driver allows:
- To bind : create a certificate of a piece of code or data, and encrypt this piece of code or data and store it in a dedicated area, if confidentiality is requested,
- To unbind : allow the verification and the decryption, if selected, of a piece of code or data be-fore to use it,
- To check a firmware certificate during Firmware run-time, in order to know if the Flash content is not corrupted,
- To provide to the application a secure mean to verify the unique association of the PLATFORM_DATA field of the manufacturer certificate and the hardware platform

Restrictions

Run-Time Checker and Run-Time Platform Data Checker services need a secure board and a secure SW build.

3.2.4.2. Secure/Public modes resource sharing

3.2.4.2.1. MKS

Purpose

The Magic Key Sequence (MKS) allows any component to define a special key sequence for an action to be triggered when this sequence is actually composed on the keypad.

The triggered action could be on the following:
- the client component is directly notified of the sequence occurrence
- extra keys are collected before being attached to the notification

Main functions

The MKS driver provides the following functions:
- add and subscribe to a magic key sequence
- remove a magic key sequence
- notify a key sequence entry

Restrictions

None

3.2.4.3. Cryptography

3.2.4.3.1. HASH

Purpose

The driver offers services for data hashing using the SHA1/MD5 security hardware module of the Ca-lypso+ chipset.

The SHA1/MD5 security module provides hardware accelerated hash functions. It can run either the SHA-1 algorithm in compliance with FIPS 180-1 standard or the MD5 message-digest algorithm developed by Rivest in 1991. Up to 2^27 bytes (128 Mbytes) of data can be hashed in a single operation and will produce a 160-bit signature in the case of SHA-1, and a 128-bit signature in the case of MD5.

Main functions

The HASH driver allows:
- To initialize either a SHA-1 or a MD5 context,
- Hash data,
- set the DMA mode for copying the hash data to the hardware FIFO

Restrictions

1. The Message to hash must be smaller than 2^35 bits (2^32 bytes or 4.29 Gigabytes) while FIPS 180-1 allows up to 2^64 bits message.
2. The message has to be a multiple of bytes. FIPS 180-1 does not necessarily require messages to be a multiple of bytes, size is specified in bits up to 2^64 bits. For messages that are not multiples of bytes, the padding to the byte boundary would have to be done by software first in order to get a message a multiple of byte. Then the hash can be hardware accelerated by the SHA1MD5 module.

TO BE CHECK

3.2.4.3.2. CRY

Purpose
The driver offers services for data encryption/decryption using the DES/3DES security hardware module of the Calypso+ chipset. Both DES and 3DES are supported in compliance with FIPS 46-3 standard. It supports ECB (Electronic Codebook) and CBC (Cipher Block Chaining) modes of operation. It does not support the CFB (Cipher Feedback) and the OFB (Output Feedback) modes of opera-tion in hardware.

Main functions
The CRY driver allows:
- To create a new DES-key object which can be used for DES or 3DES operations,
- To retrieve the length and key value components from a key object,
- To initialize a new DES-context which contains all necessary information for the encryption/decryption
- To encrypt,
- To decrypt

Restrictions
None

3.2.4.3.3. RNG

Purpose

The RNG (Random Number Generator) module is a hardware device within the Calypso+ chipset. This hardware module generates a 32 bit random number; this number is not deterministic and com-plies with FIPS-140-1 standard.

Main functions

The RNG driver allows to:
- return a 32 bit random number

Restrictions

The hardware requires a large number of clock cycles to initialise after the power ON. Refer to RNG specification for more details.

3.2.4.3.4. MPK

Purpose

The Manufacturer Public Key driver (MPK) provides a service to read the Manufacturer Public Key register of the Calypso+ chipset. The MPK-identifier is a 128-bit register composed of fuse cells electri-cally programmed and enclosing the 128 lsb of the hashing value (SHA-1) of the Public Key compo-nent of the manufacturer's Public Key pair. This value is used to authenticate any certificate signed with the Private Key from the Manufacturer.

Main functions

The MPK driver allows to:
- return the Manufacturer Public Key identifier value

3.2.4.3.5. GEA

Purpose

The GPRS Encryption Algorithm Driver is used by the moden for cyphering features. Locosto need the GEA 1/2/3.

Main functions

Restrictions

SW Component: GEA_

3.2.5. Imaging_domain

Purpose

The Imaging domain contains all SW related to picture and video features.

Main functions

The Locosto platform support still image feature using a camera sensor or still picture and video with GoldenEye companion chip.

See DAD Imaging and DAD Camera.

Restrictions

3.2.5.1. GoldenEye

Purpose

Refer to DAD Camera/GoldenEye

3.2.5.2. Camera

Purpose

Refer to DAD Camera/GoldenEye

3.2.5.3. JPEG Coedc

3.2.5.4. GIF Decode

3.2.5.5. Video

3.2.5.6. CAMD

Purpose

The Camera Driver offers straightforward access to the Calypso+ camera module. To this end, the CAMD SWE uses services offered by the lower level drivers to provide
- camera sensor control (e.g. image size, gamma correction)
- data acquisition (viewfinder frames, snapshots)

The camera is driven through the I2C module, data are coming through the PPI and transferred thanks to the DMA module.

Main functions

The CAMD driver allows to:
- enable/disable the sensor,
- set and get all snapshot parameters,
- capture a snapshot image and store it,
- set and get all viewfinder parameters,
- enter viewfinder mode

Restrictions

This driver currently supports the Agilent ADCM-2700 Camera sensor, which is the camera module of the daughter board of the E-Sample reference design board.

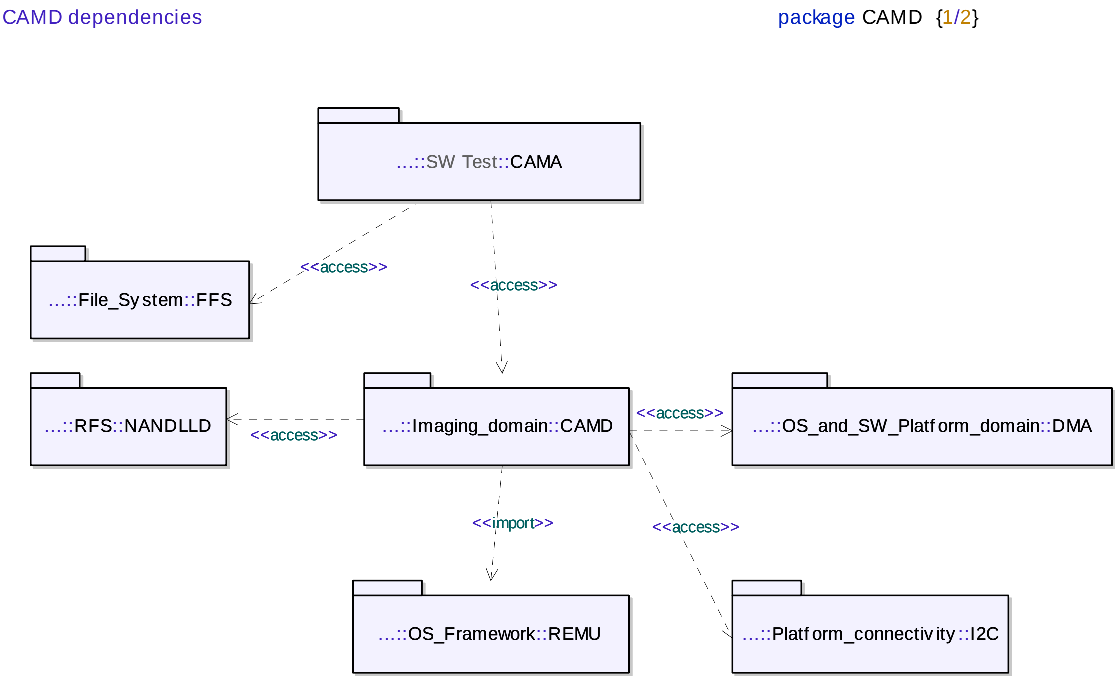


Figure 33. CAMD dependencies (Package Diagram)

3.2.6. Audio_and_Speech_domain

Audio & Speecj dependencies diagram

AudioServices and Ringer uses Beatnik engine.
AudioServices call MP3 for MP3 features

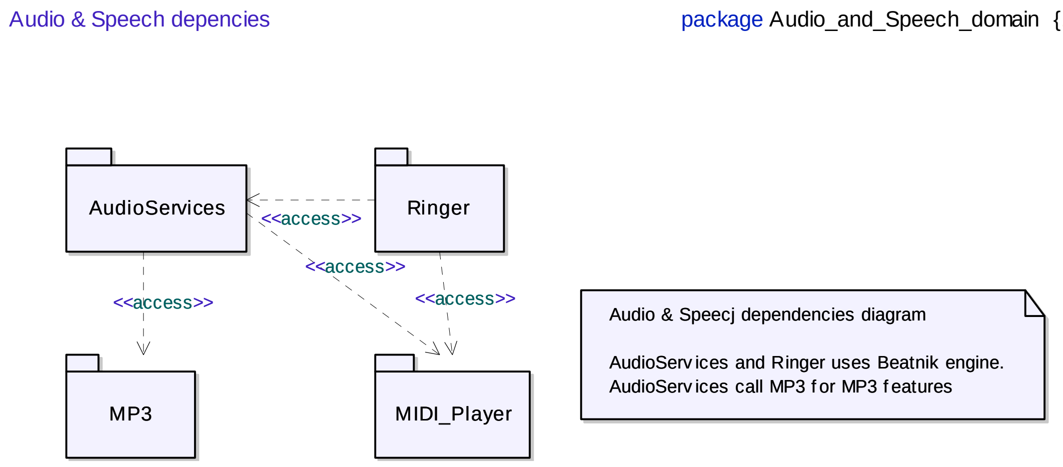

Audio & Speech depencies
package Audio_and_Speech_domain {1/1}
AudioServices
Ringer
MP3
MIDI_Player
<<access>>
<<access>>
<<access>>
<<access>>
Audio & Speecj dependencies diagram
AudioServices and Ringer uses Beatnik engine.
AudioServices call MP3 for MP3 features

Figure 34. Audio & Speech depencies (Package Diagram)

3.2.6.1. AudioServices

Purpose

The AUDIO services provided by the AUDIO features for the Locosto platform: key beep, tones, Voice memo, Speech recognition, AMR .... format.

Main functions
The audio service provides the following functions:
+ key beep generation
+ tones generation
+ Voice memorization
+ Speech recognition
+ AMR
+ MMS
+ Audio configuration

Restrictions
The melody format can only play 8 notes in parallel instead of 16, due to DSP CPU load constraints.

SW component: audio

3.2.6.2. MIDI_Player

Purpose

The MobileBAE (Beatnik Audio Engine) is a machine independent, audio device independent, soft-ware engine for music and sound playback, incorporating a music synthesizer and advanced interac-tivity features, and designed specifically for mobile devices.

The ‘wrapper’ code is the C language interface that makes mobileBAE ’s machine independence pos-sible by placing a platform abstraction layer (PAL) at the boundary between the mobileBAE core and its host platform (OS services and audio hardware).

The mobileBAE for Riviera/Calypso+ is a mobileBAE integrated package containing all the source and binaries required to build, evaluate and distribute the mobileBAE audio engine running on Calypso+ hosting the Riviera platform.

Main functions
The input formats supported are:
- MIDI file
- IMelody Monotone
- Extensible Music Format (XMF)
- RMF
- SMAF

Restrictions
None

To check: Support play from file and play from memory. Use case : ring tone for ULC.

3.2.6.3. MP3

Purpose

The MP3 features are handled in the DSP core.

Main functions

To be completed

Restrictions

3.2.6.4. Ringer

Purpose

The MIDI Ringer package handles the ringer feature of the Locosto platform inhereted of the TCS 3.1.

Main functions

It support the following features:
- three ringer mode: incoming call, incoming SMS and Alarm
- settings for each ringer mode
- dynamic update voices according to the platform activity
  + 16 voices during GPRS activity
  + 24 voices during incoming cal
  + 32 voices for playing MIDI files

Restrictions

3.2.6.5. TTY

Purpose
Set the tty audio profile for the CTM mode

Main functions

To be completed

Restrictions

3.2.7. Connectivity_domain

Purpose

The connectivity domain contains all package used for the connection of the device with external devices.
Connectivity dependencies diagram

The connectivity domain contains USB, BlueTooth,
IRDA and UART drivers.

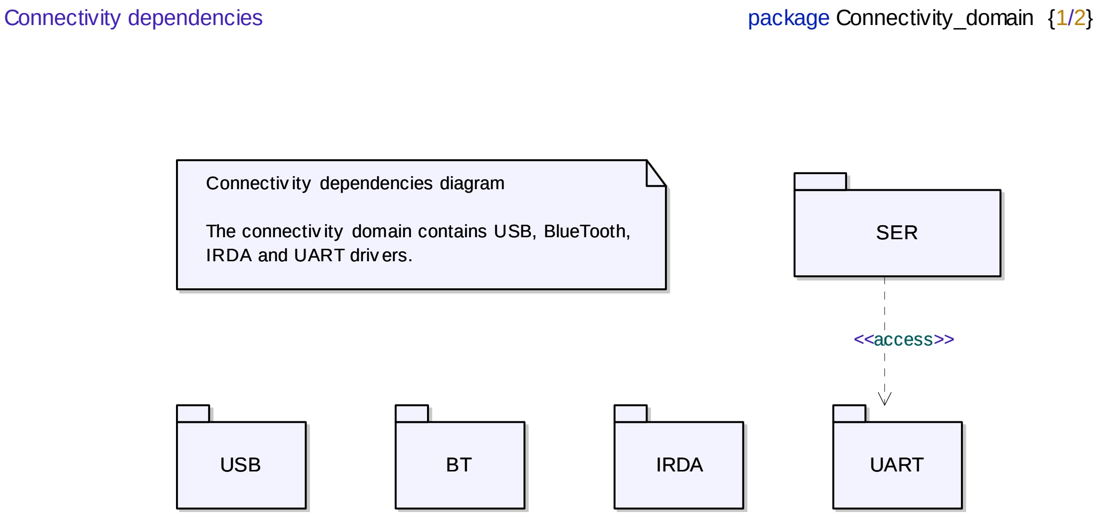

Connectivity dependencies
package Connectivity_domain {1/2}
Connectivity dependencies diagram
The connectivity domain contains USB, BlueTooth, IRDA and UART drivers.
SER
<<access>>
USB
BT
IRDA
UART

Figure 35. Connectivity dependencies (Package Diagram)

3.2.7.1. USB

Purpose

The USB LoCosto module supports the implementation of a "Full-Speed" device fully compliant to USB 1.1 standard. It provides an interface between a local host (LH), i.e the ARM processor and the USB wire.The module is not "on the go".

Main functions

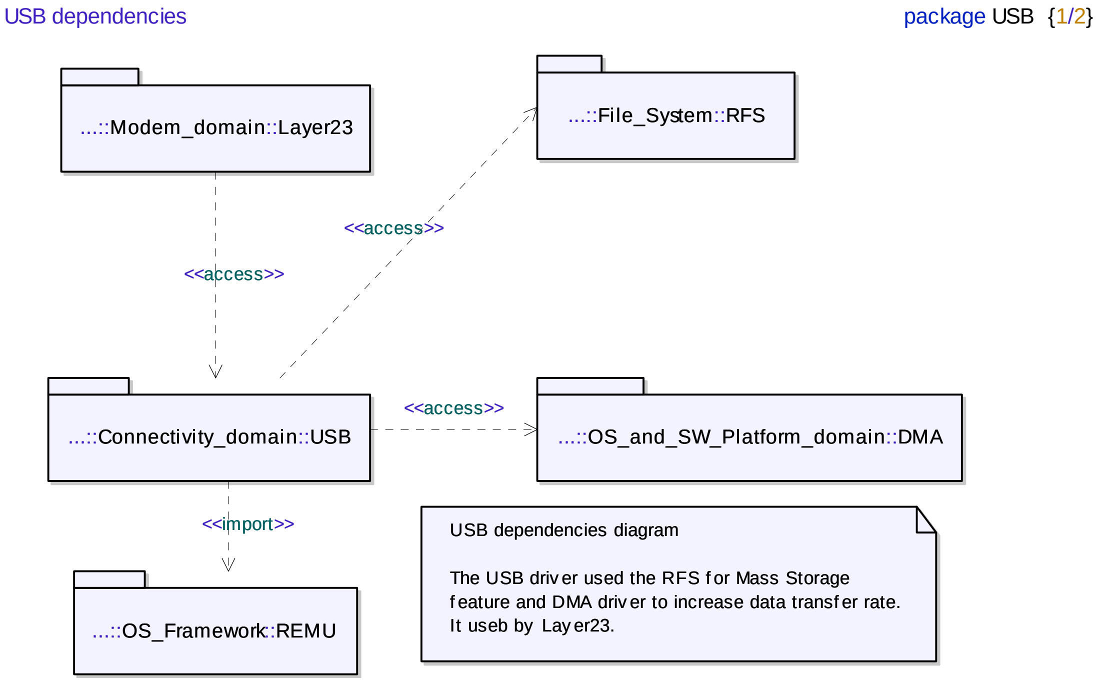
The USB driver allows:
- Support USB 1.1 standard requests
- a FM to subscribe an interface,
- a FM to unsubscribe an interface,
- to retrieve information about endpoint associated with the given interface (status),
- sent data to the host,
- receive data from the host

Restrictions
None

See charging USB. Perfromances: no internal memory.

USB dependencies diagram

The USB driver used the RFS for Mass Storage
feature and DMA driver to increase data transfer rate.
It useb by Layer23.



USB dependencies diagram

The USB driver used the RFS for Mass Storage
feature and DMA driver to increase data transfer rate.
It useb by Layer23.

Figure 36. USB dependencies (Package Diagram)

3.2.7.2. BT

Purpose

Refer to DAD BlueTooth.

3.2.7.3. UART

Purpose
See "Serial Dynamic Switch - Purpose" section

Restrictions

The UART IrDA cannot be use for Fax & Data flow since the DCD and DTR signals needed for the right behavior of the driver are only available on UART Modem. Fax & Data flow could be enabled by using a Software flow control mechanism, but this is not currently supported.
- The Fax&Data flow cannot be output on the USB.

TO BE CHECK: perhaps remove is use USB.

3.2.7.4. IRDA

Purpose

The Infra RED features..

3.2.7.5. CarKit

3.2.7.6. Ethernet

3.2.7.7. Headset

3.2.7.8. SER

Purpose

The E-Sample board offers two UART serial ports amongst the three available on Calypso+, and a USB port.

The Serial Dynamic Switch (SER) allows managing the usage of these two serial ports provided by the E-Sample board: the UART Modem and the UART IrDA, and the USB port.

It allows the platform to manage trace flows, Fax & Data flows. This component presents a unique interface to the serial port that manages data flows according to its configuration.

Main functions

The Serial Dynamic Switch driver provides the following facilities:
+ Set up, read status and wake up
+ Communication (trace) functions (call UART driver API or USB driver API)
+ Fax & data functions (call UART Fax & Data driver)

Restrictions

The USB port can output only the Trace flow.

Locosto: 1 UART, perhaps need reowkr and study for traces, multiplexing and AT command...
SER Dependencies diagram

The SER is used to access to tthe UART:
- by SAM for PS access

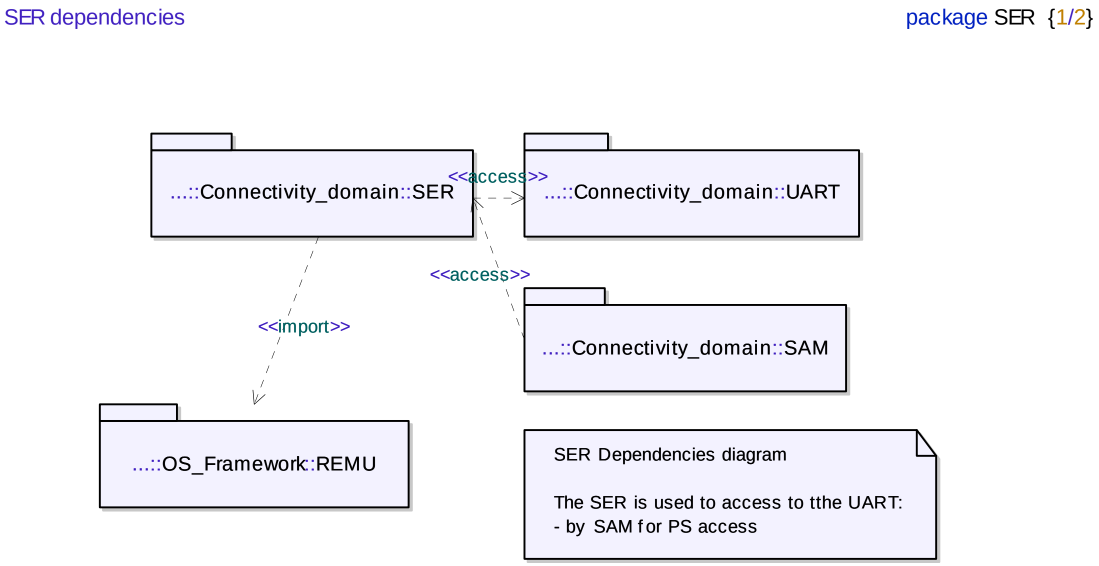


Figure 37. SER dependencies (Package Diagram)

3.2.7.9. SAM

Purpose

To be completed: need information from TI Berlin

Main Function

Restriction

3.2.7.10. Multiplexer

Purpose

To be completed: need information from TI Berlin

Main Function

Restriction

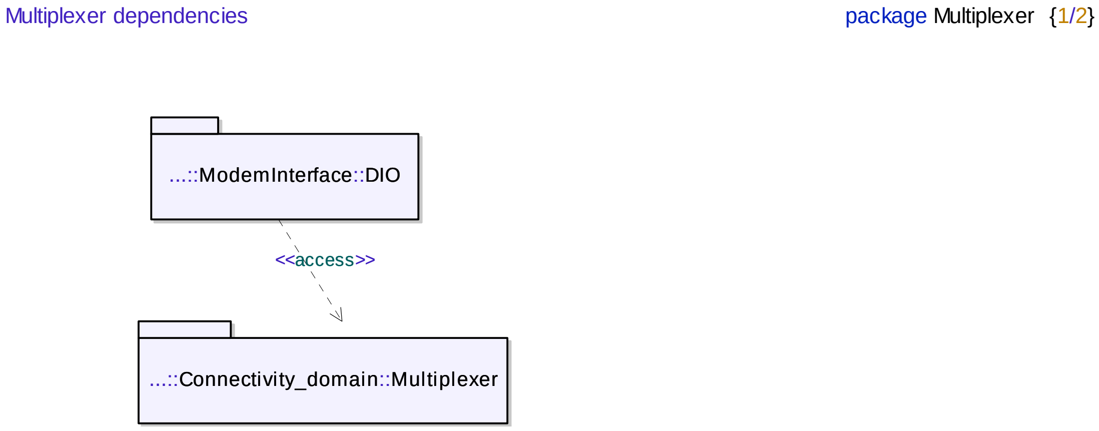

Multiplexer dependencies
package Multiplexer {1/2}
...::ModemInterface::DIO
<<access>>
...::Connectivity_domain::Multiplexer

Figure 38. Multiplexer dependencies (Package Diagram)

3.2.7.11. Accesory_Detection

3.2.8. Application_Java_and_UI_domain

Purpose

The Application, Java and UI domain is not addressed in this document.

3.2.8.1. Audio_and_Speech

3.2.8.1.1. MP3

The MP3 Player functional Block provides the following MP3 features to the end-user:
- play MP3 files
- stop playing MP3 file
- pause the MP3 player
- resume the MP3 player

3.2.9. User_Interactions_domain

3.2.9.1. LCD

Purpose

The LCD driver provides function to display "Framebuffer" in the screen.

LCD Driver dependencies diagram

This diagram shows the internal dependencies of the
the LCD driver.

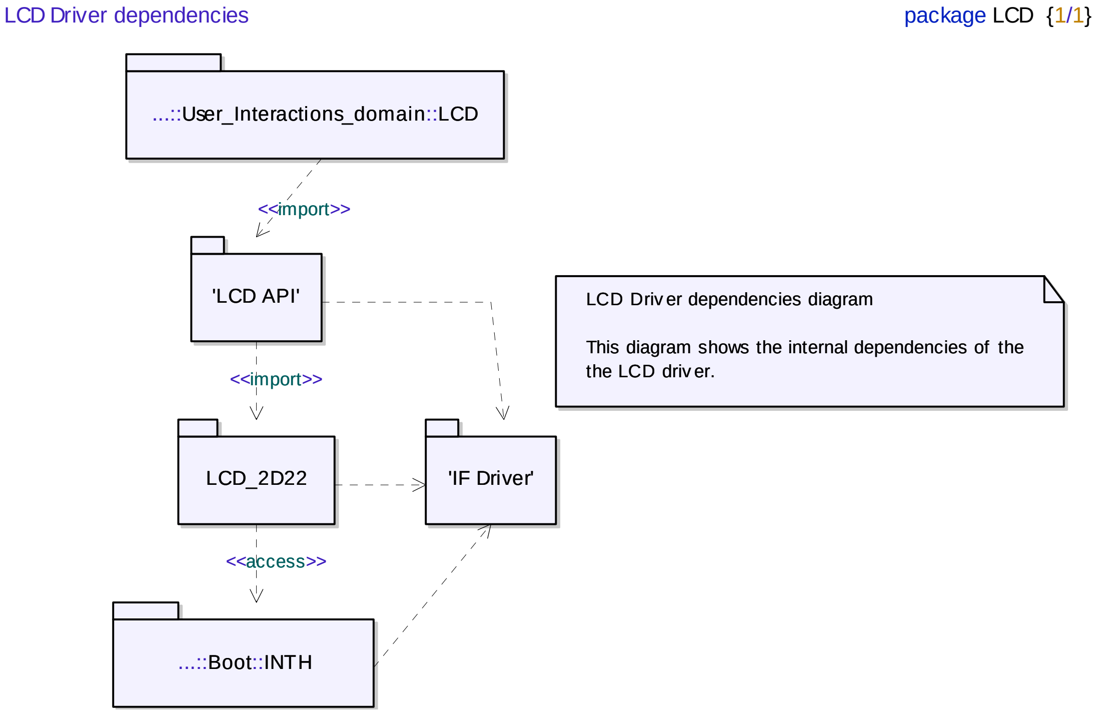


Figure 39. LCD Driver dependencies (Package Diagram)

3.2.9.1.1. LCD API

Purpose

The LCD interface provides an interface to the MCU to load data from internal SRAM into the Display RAM of an external LCD controller. This interface is used also to update configuration registers, to read status information and graphical data from LCD controller.

Main functions

Restrictions

3.2.9.1.2. LCD_2D22

Purpose

This package contains the implementation of the current LCD low level driver based on the current sensor.

3.2.9.1.3. IF DRIVER

Purpose

This package contains the interruption handling between the LCD driver and the driver packages.

3.2.9.2. KPD

Purpose

The main purpose of the keypad driver (KPD) is to send messages to registered SWE when keys are pressed or released on the mobile. It is based on client/server mechanism with subscription operation.

The driver manages two types of key concept:
- physical keys: define the set of keys available on the physical keypad
- virtual keys: abstraction of physical key to limit the impact of keypad configuration

A subscriber has no visibility on the physical keys. It only uses virtual key ID for all the keypad ser-vices.

Main functions

The keypad driver allows to:
- subscribe a SWE to a predefined set of key; when one of these keys is pressed or released, the SWE is automatically notified
- unsubscribe from keypad (for all keys)
- define repetition for a key (long press and repetition available for a specific subscriber)
- define a keypad owner; this subscriber is then the only one notified for key events until it does unsubscribe or cancels this privilege
- define several modes (mappings) for associating physical key and functional key
- switch from a mode to another
- map characters to keys in default or alphanumeric mode

Note:
The time resolution can be higher if the hardware decoding mode is used. This mode can be enabled using a compiler switch.

Some more functionalities are available in this mode. It is then possible for a SWE to:
- receive a notification when another subscriber has changed the timer values for long press and repetition
- receive a notification when a long time of inactivity of the keypad is detected and a time out has occurred
- receive a notification when a keypad miss event has been detected.

KPD dependencies diagram

The KPD is used only by Application and UI domain
or Test and Debug domain.

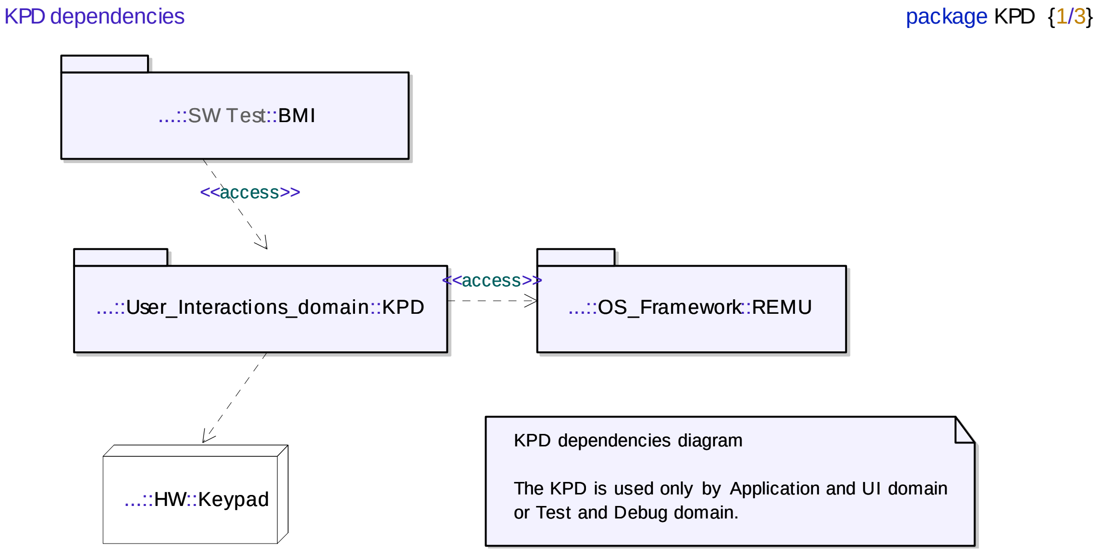


Figure 40. KPD dependencies (Package Diagram)

3.2.9.3. Lights

Purpose

This package contains all light/backlight features and associted sotwate components.

3.2.9.3.1. LLS

Purpose
The Low Level Services (LLS) SWE allows access to some hardware interfaces.

Main functions
The functions provide an on/off switch mechanism to drive one of the three LEDs: LED A, backlight and pre-charge LED.

Restrictions

The current release provides functions only for the LED management.

3.2.9.4. Buzzer

Purpose

The Buzzer provides functions to start/stop the buzzer.

Main functions

Start/Stop buzzer.
This features should be disable using configuration/compilation parameter

Restrictions
None.

3.2.10. Test_and_Tools_domain

Purpose

This domain contains all SW components define for test purpose and tools to debug, test, configure and customize the platform.
Test and Tools dependencies diagram

This diagram shows how BMI deals with the Modem.
BMI and MFW targets only anchor 1.

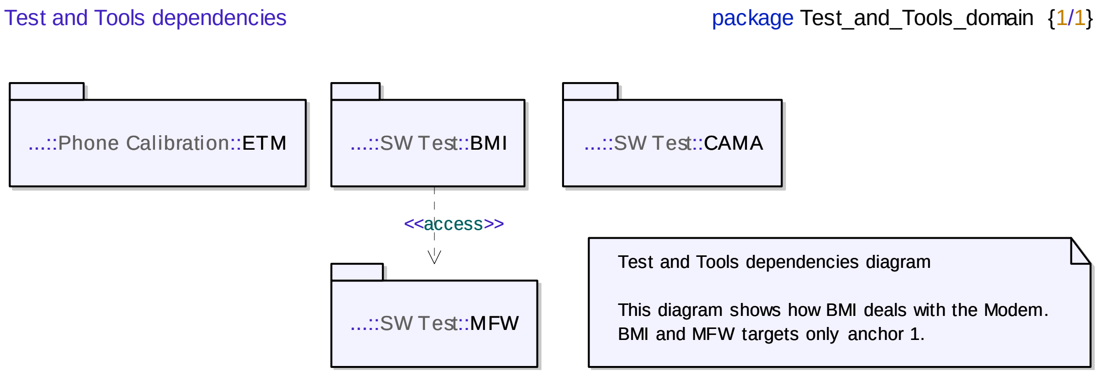

Test and Tools dependencies
package Test_and_Tools_domain {1/1}
...::Phone Calibration::ETM
...::SW Test::BMI
...::SW Test::CAMA
<<access>>
...::SW Test::MFW
Test and Tools dependencies diagram
This diagram shows how BMI deals with the Modem.
BMI and MFW targets only anchor 1.

Figure 41. Test and Tools dependencies (Package Diagram)

3.2.10.1. SW Test

3.2.10.1.1. BMI

Purpose

The Basic Man Machine Interface (BMI) provides a test user interface for the call management, SIM management, phonebook management, SMS and phone settings.

Main functions

The BMI or Basic MMI is used for the menu screen presentation. It is divided into a number of modules.

Standard Modules
- Phonebook
- Call
- Pins
- Supplementary Services (SS)
- Short Message Services (SMS)
- Network Service

Common Components

- Menus
- Editors
- Languages
- Dialogue
- List
- Icons

It is integrated with Ezitext to support Chinese and Predictive text

Restrictions

TCS3.1 BMI does not support WAP, MMS, Bluetooth, Java

Test and Tools dependencies diagram

This diagram shows how BMI deals with the Modem.
BMI and MFW targets only anchor 1.

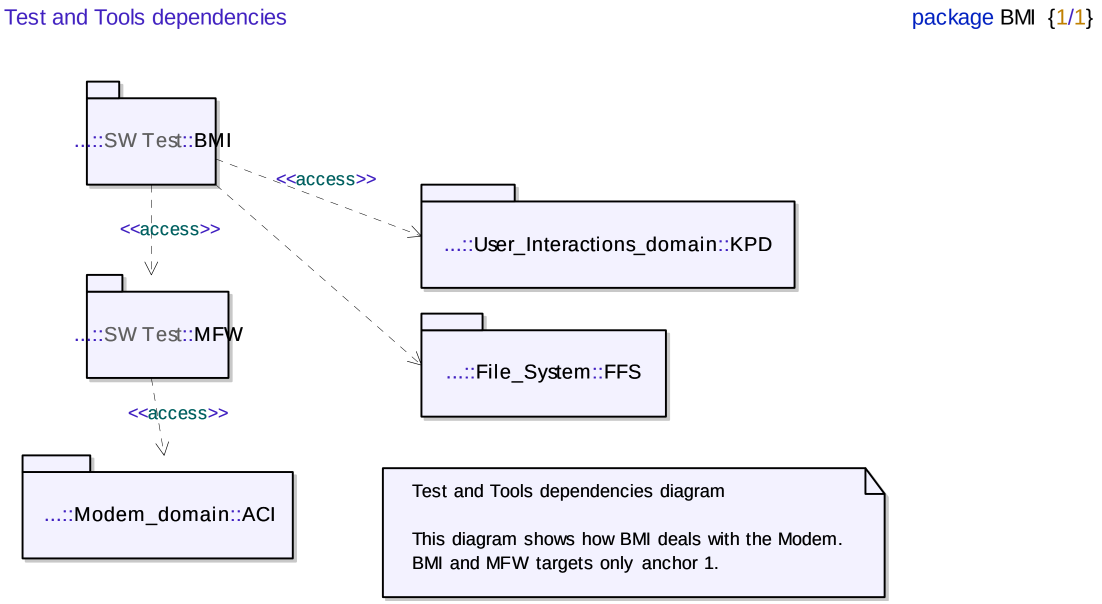

Test and Tools dependencies
package BMI {1/1}
...::SW Test::BMI
<<access>>
<<access>>
...::User_Interactions_domain::KPD
...::SW Test::MFW
...::File_System::FFS
<<access>>
...::Modem_domain::ACI
Test and Tools dependencies diagram
This diagram shows how BMI deals with the Modem.
BMI and MFW targets only anchor 1.

Figure 42. Test and Tools dependencies (Package Diagram)

3.2.10.1.2. MFW

Purpose

The Mobile Framework provides communication layer and API to access to the ACI layer from the BMI.

Main function

It is divided into a number of modules which use common components in their presentation.

Standard Modules

- SIM
- Phonebook
- Call
- Supplementary Services (SS)
- Short Message Sequences (SMS)
- Common Components

Window

- Keypad
- Timers
- Menus
- Editors
- Icons

Additional Modules

- General Packet Radio Resources (GPRS)
- CPHS

Restrictions

TCS3.1 BMI does not support WAP, MMS, Bluetooth, Java

3.2.10.1.3. CAMA

Purpose
The Riviera Camera application (CAMA) provides viewfinder and snapshot capturing services. If CAMA is enabled, the LCD displays the viewfinder. The snapshot services allow taking snapshots images. The snapshot image is displayed on the LCD and can be returned to the clients in both YUV and JPEG format.

The picture below shows the dependencies of CAMA towards other software entities. CAMA is using a camera driver (CAMD) to read images from the camera sensor. The IMG-library is used for JPEG en-coding. The LCD-driver is used to display the viewfinder and snapshot images on the LCD.
The RFS library is used to write the image to a file.

Main functions
Same as CAMD, plus:
- encode the YUYV422 image into a JPEG image,
- save the image to the given directory path

Restrictions
None

3.2.10.1.4. DBG

Purpose

The Debug Module (DBG) allows to map pins on the outpin spin for debug.

Main function

To be completed

Restriction
DBG dependencies diagram

To be completed

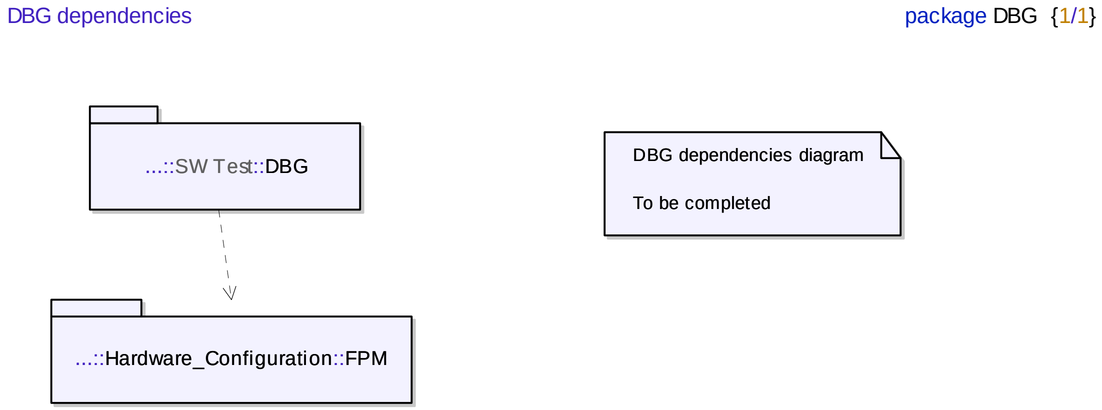


Figure 43. DBG dependencies (Package Diagram)

3.2.10.2. Crash Diagnose

3.2.10.2.1. DAR

Purpose

The Diagnose and Recovery component (DAR) provides an easy way for developers, integrators and testers to access to information for anomalies, sent by the involved software components. It also pro-vides a recovery mechanism that allows restarting the system (automatically or manually) in a clean and transparent way.

Main Functions

DAR is a SWE running and collecting data from the embedded modules. It handles two types of data:
+ Symptom perception: first filtering attribute
  - Freeze Category: it seems nothing is running in the handset
  - Reset Category: the mobile re-starts abnormally
  - One function systematically didn’t work
  - One function erratically (sometimes) didn’t work
+ Exceptions: generated by internal or external sources to cause the processor to handle event
  - Undefined instruction: when the coprocessor didn’t execute the called instruction
  - Pre-fetch abort: signaled by the memory system when fetch instruction is invalid
  - Data abort: signaled by the memory system when access to data is invalid

All information contains a level:
- DAR_ERROR: when error is detected and reveals that the system not working correctly
- DAR_WARNING: when error is detected but handles by the system
- DAR_DEBUG: debug information
- DAR_NO_DIAGNOSE: the SWE doesn’t want the diagnose information about another SWE

The main functions are:
- get recovery status (from MMI)
- recovery configuration
- get recovery data
- start/stop/reload watchdog timer
- reset system
- set diagnose filter
- write diagnose record
- store diagnose in RAM buffer when an emergency has been detected

Restrictions
None

DAR dependencies diagram

The DAR driver is based on RVF frame and
uses FFS to store recovery data.

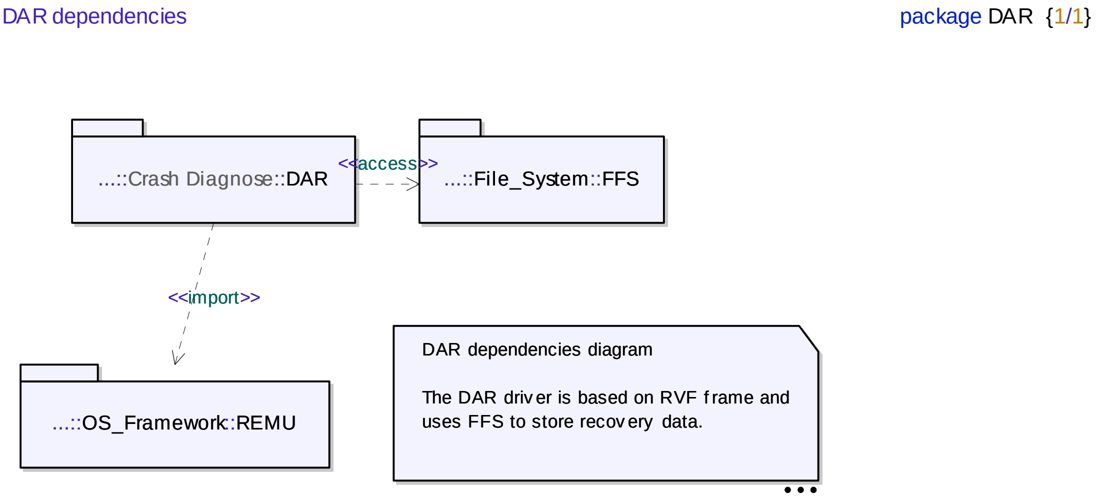


Figure 44. DAR dependencies (Package Diagram)

3.2.10.3. Flash Loader

Purpose

The Flash Loader alows to flash the image of the code on the target according to secuirty features. It sall be done using USB link.

3.2.10.4. SW Development

Purpose

This section should contains a desciption of all the tools use for development as IDE, debugge...

3.2.10.4.1. Windows

Purpose

The Windows packages is used for development and test in a Windows environment.

Main functions

Restrictions

3.2.10.4.2. Nucleus_MNT

Purpose

This package contains a implementation of Nucleus for windows. It allows to run all GPF entity under windows.

3.2.10.5. Phone Calibration

Purpose

This section describes the SW component involved for the Phone Calibration.

3.2.10.5.1. ETM

Purpose

The Enhanced Test Mode (ETM) is set of components used to provide a test environment (ETM Sys-tem) for the Riviera platform. It uses the UART driver to communicate between the PC and the target. A shell command (tmsh) is used for debugging and setup of the test mode.

Main functions

Each module to test on the target has a corresponding module in the PC. During the test, they ex-change data through ETM module and Test Mode Transport module (TMT). The communication pro-tocol is independent of the physical transport medium.

It perfroms via the API the following functions :
- issue AT commands towards ACI
- activate audio functionalities in order to test them
- Address the ABB chip directly
- Read the DIE-Id

The ETM/TMT interface provides functions to exchange data with the PC/target module:
- copy module identifier in the packet
- put/get data in a packet (8bit, 16bit or 32 bit)
- put buffer in a packet
- send/receive packet
- free packet

Restrictions

None

ETM dependencies diagram

The ETM software used ACI and ABB features in order to
Calibrate the DRP and additional data.

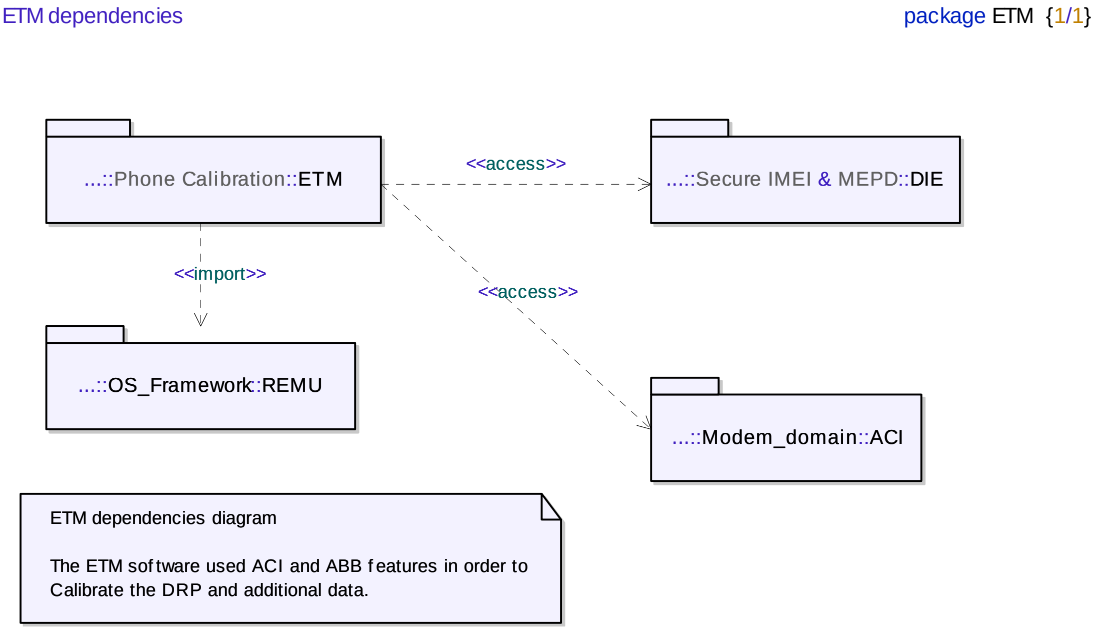

ETM dependencies
package ETM {1/1}
...::Phone Calibration::ETM
<<access>>
...::Secure IMEI & MEPD::DIE
<<import>>
<<access>>
...::OS_Framework::REMU
...::Modem_domain::ACI
ETM dependencies diagram
The ETM software used ACI and ABB features in order to Calibrate the DRP and additional data.

Figure 45. ETM dependencies (Package Diagram)

3.2.10.6. System Debug

Purpose

This section should described all sW component required for debug purpose of the Locosto board.

3.2.10.7. SW Trace

Purpose

This section should described all SW component involved for trace purpose.

3.3. Critical Use cases study

In order to find a good trade-off between CPU load and usage of internal memory the following use cases have been identified. These use cases represents the worst case in term of CPU load requirements. In order to reduce the CPU load the functional blocks could execute critical parts of their code from internal RAM or/and access their static data from there. The complete set of use cases for a phone is not part of this document.

The use cases describe the maximum parallel activities. The CPU load investigation will show that some of them are not applicable on this platform and need to be changed or removed completely.

R 1.1.1. #1 Accept Mobile Terminated voice call, phone is in idle state

State: Phone runs in idle mode. Camped on a cell while observing neighbor cells. Phone monitors the paging channel. Possibly reading and decoding CBCH.
Event: Receives a paging message from the network
Modem answers the paging and change into dedicated mode. In parallel a MIDI ringing melody is started and the idle screen changes to a ringing indicator.

R 1.1.2. #2 Voice Memo record/playback while voice call

State: Modem is in voice call (TCH assigned). Neighbor cells are monitored in parallel. AMR Voice codec and possibly the CTM codec are active. LCD shows call status.
Event: User starts voice memo record or playback.
Modem performs AMR voice coding and stores data into FFS or Modem retrieves data from FFS and perform AMR voice decoding. LCD shows status of voice memo (playback/record status)

R 1.1.3. #3 SMS Reception/MMS Notification while in dedicated mode

State: Modem is in voice call (TCH assigned). Neighbor cells are monitored in parallel. AMR Voice codec and possibly the CTM codec are active. LCD shows call status.
Event: Modem receives a SMS
Modem decodes the SMS and forwards it towards SMBS and to the application. The LCD shows the reception of an SMS/MMS.

R 1.1.4. #4 Voice call MO/MT with Bluetooth connection

State: Modem is in voice call (TCH assigned). Neighbor cells are monitored in parallel. AMR Voice codec and possibly the CTM codec are active. LCD shows call status. Bluetooth headset is connected via radio link to the modem.
Event: none
Dedicated mode and Bluetooth stack is active in voice profile where speech data are transferred via MCSI towards the voice codec running on the DSP.

R 1.1.5. #5 Upload/download MMS via CSD

State: Modem is in a non-transparent circuit switched data connection with assigned TCH. Neighbor cells are monitored in parallel. User has started the download of an MMS from the server. Status of download is displayed in the LCD.
Event: none
Modem performs a CSD transfer with PPP active and has to store the MMS in the file system (NAND/NOR Flash)

R 1.1.6. #6 Upload/download MMS via GPRS

State: Modem performs GPRS packet transfer. PDP context is activated. User has started the download of an MMS from the server. Status of download is displayed in the LCD.
Event: none
Modem performs UL/DL TBF and forward IP packets from/to the TCP/IP stack of the application suite. Received MMS need to be stored in the file system (NAND/NOR Flash)

R 1.1.7. #7 Java application execution while voice call via Bluetooth

State: Modem is in voice call (TCH assigned). Neighbor cells are monitored in parallel. AMR Voice codec and possibly the CTM codec are active. LCD shows call status. Bluetooth headset is connected via radio link to the modem.
Event: User starts a application which runs on a JAVA VM.
Dedicated mode and Bluetooth stack is active with voice profile where speech data are transferred via MCSI towards the voice codec running on the DSP. JAVA VM performs decoding and execution of JAVA code (e.g. a complete MMI or a application like a game with a certain LCD frame rate)

R 1.1.8. #8 Java application execution with MIDI sound and CSD data connection

State: Modem is in a non-transparent circuit switched data connection with assigned TCH. Neighbor cells are monitored in parallel. Download of an MMS from the server is active. Status of download is displayed in the LCD.
Event: User starts a JAVA based game which has a MIDI based sound background.
Modem performs a CSD transfer with PPP and TCP/IP active and has to store the MMS in the file system (NAND/NOR Flash). JAVA VM performs decoding and execution of JAVA code of the game and the MIDI player is active in order to play the sound background.

R 1.1.9. #9 Java application execution with MIDI sound and GPRS data connection via Bluetooth

State: PDP context is activated. Dial-up connection from a PC via Bluetooth is active.
Event: User start a JAVA based game which has a MIDI based sound background.
JAVA VM performs decoding and execution of JAVA code of the game and the MIDI player is active in order to play the sound background. In parallel the modem performs UL/DL TBF and forward IP packets from/to the Bluetooth stack which exchange the data with a PC over the Bluetooth radio link. The TCP/IP of the phone is not running since it is running in the PC itself.

R 1.1.10. #10 Java application without MIDI sound while voice call via Bluetooth

State: Modem is in voice call (TCH assigned). Neighbor cells are monitored in parallel. AMR Voice codec is active. LCD shows call status. Bluetooth headset is connected via radio link to the modem.
Event: User starts a JAVA application without sound like a calculator, calendar or a game
Dedicated mode and Bluetooth stack is active in voice profile where speech data are transferred via MCSI towards the voice codec running on the DSP. Bluetooth stack is active with voice profile where speech data are transferred via MCSI towards the voice codec running on the DSP. JAVA VM performs decoding and execution of JAVA code. LCD is updated with a certain frame rate.

R 1.1.11. #11 Camera view finder in Idle Mode

Initial State: Phone runs in idle mode. Phone camped on a cell while observing neighbor cells. The phone monitors the paging channel. Possibly reading and decoding CBCH.
Event: User starts the camera application in view finder mode
Modem performs monitoring of the serving and neighbor cells. CBCH decoding is possible. Also cell-reselection could be performed. Camera application has been started. After the camera sensor has been setup the camera driver programs the DMA in order to transfer the preview picture QCIF into the LCD frame-buffer memory which is located in the external RAM with a rate of 15 frames per second. With the same rate a second DMA transfers the frame-buffer content to the LCD.

R 1.1.12. #12 Camera snapshot in Idle Mode

State: Phone runs in idle mode. Phone camped on a cell while observing neighbor cells. Phone monitors the paging channel. Possibly reading and decoding CBCH. Camera application and driver are in preview mode two DMA are utilized to transfer preview frames from the sensor to the frame buffer and further towards the LCD.
Event: User activates the camera snapshot function by pressing a button
Idle mode is continuously maintained by the modem. Preview mode is stopped. In parallel the sensor is programmed to acquire a VGA snapshot. Snapshot data need to be transferred by DMA to the external RAM. In parallel a shutter sound needs to be played to indicate the acquisition to the user. Afterwards the picture is compressed with the JPEG encoder and stored together with a generated thumbnail into the file system (NOR/NAND flash).

R 1.1.13. #13 Camera view finder while voice call

State: Modem is in voice call (TCH assigned). Neighbor cells are monitored in parallel. AMR Voice codec and possibly the CTM codec are activated. LCD shows call status.
Event: User starts the camera application in view finder mode
Voice call is continued. Camera application has been started. After the camera sensor has been setup the camera driver programs the DMA in order to transfer the preview picture QCIF into the LCD frame-buffer memory which is located in the external RAM with a rate of 15 frames per second. With the same rate a second DMA transfers the frame-buffer content to the LCD.

R 1.1.14. #14 Camera snapshot while voice call

State: Modem is in voice call (TCH assigned). Neighbor cells are monitored in parallel. AMR Voice codec and possibly the CTM codec is activated. LCD shows call status. Camera application and driver are in preview mode two DMA are utilized to transfer preview frames from the sensor to the frame buffer and further towards the LCD.
Event: User activates the camera snapshot function by pressing a button
Voice call is continued. Preview mode is stopped. In parallel the sensor is programmed to acquire a VGA snapshot. Snapshot data need to be transferred by DMA to the external RAM. In parallel a shutter sound needs to be played to indicate the acquisition to the user. Afterwards the picture is compressed with the JPEG encoder and stored together with a generated thumbnail into the file system (NOR/NAND flash).

R 1.1.15. #15 Camera view finder while GPRS data transfer

State: Modem is in packet transfer mode on the serving cell. Neighbor cells are monitored in parallel. Either the internal TCP/IP stack is connected to the GPRS modem, serving a application like MMS download or the GPRS modem is connected via USB to a PC where in this case the host performs TCP/IP and e.g. web-browsing.
Event: User starts the camera application in view finder mode
Modem continues the UL/DL TBF. Camera application has been started. After the camera sensor has been setup the camera driver programs the DMA in order to transfer the preview picture QCIF into the LCD frame-buffer memory which is located in the external RAM with a rate of 15 frames per second. With the same rate a second DMA transfers the frame-buffer content to the LCD.

R 1.1.16. #16 Camera snapshot while GPRS data transfer

State: Modem is in packet transfer mode on the serving cell. Neighbor cells are monitored in parallel. Either the internal TCP/IP stack is connected to the GPRS modem, serving a application like MMS download or the GPRS modem is connected via USB to a PC where in this case the host performs TCP/IP and e.g. web-browsing. Camera application and driver are in preview mode two DMA are utilized to transfer preview frames from the sensor to the frame buffer and further towards the LCD.
Event: User activates the camera snapshot function by pressing a button
Modem continues the UL/DL TBF. Preview mode is stopped. In parallel the sensor is programmed to acquire a VGA snapshot. Snapshot data need to be transferred by DMA to the external RAM. In parallel a shutter sound needs to be played to indicate the acquisition to the user. Afterwards the picture is compressed with the JPEG encoder and stored together with a generated thumbnail into the file system (NOR/NAND flash).

R 1.1.17. #17 Incoming call with MIDI ringing tone while camera view finder mode

State: Phone runs in idle mode. Camped on a cell, observing neighbor cells. Monitors the paging channel. Possibly reading and decoding CBCH. Camera application and driver are in preview mode two DMA are utilized to transfer preview frames from the sensor to the frame buffer and further towards the LCD.
Event: Incoming call from the network
Modem receives the paging and reply to the network. A TCH could be assigned. One DMA is performed transferring the preview QCIF picture from the sensor into the LCD frame-buffer memory which is located in the external RAM with a rate of 15 frames per second. With the same rate a second DMA transfers the frame-buffer content to the LCD. The MIDI file which has been setup as ringing tone is loaded from file system into the external/internal RAM. BEA engine is started, preprocess the file and start playback of the MIDI content with up to 32 voices.

R 1.1.18. #18 Incoming call with MIDI while camera snapshot

State: Phone runs in idle mode. Camped on a cell observing neighbor cells. Monitors the paging channel. Possibly reading and decoding CBCH. Snapshot data need to be transferred by DMA to the external RAM. In parallel a shutter sound needs to be played to indicate the acquisition to the user. Afterwards the picture is compressed with the JPEG encoder and stored together with a generated thumbnail into the file system (NOR/NAND flash).
Event: Incoming call from the network
Modem receives the paging and reply to the network. A TCH could be assigned. Camera continues to transfer, compress and store the file from the sensor. The MIDI file which has been setup as ringing tone is loaded from file system into the external/internal RAM. BEA engine is started, preprocess the file and start playback of the MIDI content with up to 32 voices.

R 1.1.19. #19 Incoming call with MIDI while Java application is executed with MIDI sound

State: Phone runs in idle mode. Camped on a cell, observing neighbor cells. Monitors the paging channel. Possibly reading and decoding CBCH. JAVA VM performs decoding and execution of JAVA code (e.g a game) and the BAE is active in order to play the sound background.
Event: Incoming call from the network
Modem receives the paging and reply to the network. A TCH could be assigned. JAVA VM continues to execute the game. BAE engine is stopped. The MIDI file that has been setup as ringing tone is loaded from file system into the external/internal RAM. BEA engine preprocess the file and start playback of the MIDI content with up to 32 voices. A call notification on the LCD is possible.

R 1.1.20. #20 Incoming call with MIDI sound playback via Bluetooth headset connection while in idle mode

State: Modem runs in idle mode. Observing neighbor cells, while camped on a cell. It monitors the paging channel. Possibly reading and decoding CBCH. Bluetooth headset is connected via radio link to the modem.
Event: Incoming call from the network
Modem receives the paging and reply to the network. A TCH could be assigned. A Bluetooth connection is made to the headset in voice profile and the MSCI port is used to exchange voice data. The MIDI file that has been setup as ringing tone is loaded from file system into the external/internal RAM. BEA engine preprocess the file and start playback of the MIDI content with up to 32 voices via the MCSI in order to receive the melody with the headset. A call notification on the LCD is possible.

R 1.1.21. #21 Incoming call with MIDI sound playback via Bluetooth headset connection while in Java application execution (Game) playing MIDI via Bluetooth

State: Modem runs in idle mode. Observing neighbor cells while camped on a cell. It monitors the paging channel. Possibly reading and decoding CBCH. Bluetooth headset is connected via radio link to the modem and runs the voice profile connected to the MCSI. JAVA VM performs decoding and execution of JAVA code (e.g. a game) and the BAE is active in order to play the sound background via MCSI.
Event: Incoming call from the network
Modem receives the paging and reply to the network. A TCH could be assigned. The BAE which plays the games sound is stopped. The MIDI file that has been setup as ringing tone is loaded from file system into the external/internal RAM. BEA engine preprocess the file and start playback of the MIDI content with up to 32 voices via the MCSI in order to receive the melody with the headset. A call notification on the LCD is possible.

R 1.1.22. #22 – Java network game application with MIDI with Headset: PC synchronization via USB

State: Java game is running and playing MIDI as audio features with an plugged heaset

Event:

R 1.1.23. #23 – Voice call with TTY

State:

Event:

R 1.1.24. #24 - MIDI playback during background GPRS data transfer and alert generation for incoming voice call

State:

Event:

R 1.1.25. #25 - AMR-NB playback during background GPRS data transfer and alert generation for incoming voice call

State:

Event:

R 1.1.26. #26 - Camera View finder with MIDI player

State:

Event:

R 1.1.27. #27 - Camera Snap-shot with MIDI player (with 2.5S Shot-2-Shot delay)

State:

Event:

4. Tools

4.1. Software engineering tool

4.1.1. Compiler

The Locosto software should support compiler version 1.22e and 2.54 in parallel.

4.1.2. Build System

In this section are described build system enhancements:

- Runtime Library. The RTS16 lib of CGT 2.54 contains some static data which is not required in the HW environment. The optimization will require replacing the existing library by a new version.
- Use of unified BusyBe for all anchors.
- Integration of AGPS from TCS4.0 (only Anchor 4)
- Support compiler version 1.22e and 2.54 in parallel
- Integration of stubs into the build process in order to link against ROM'ed functions. Therefore a tool is required like described in [15].
- Since there are major bugs in the code generator of the CGT 2.54 most probably the 2.55 version need to be applied for LoCosto.

4.1.3. Profiler

TI profiler should be use to perform performances measurements.

4.1.4. Debugger

Current TCS 3.1 debugger.

This page intentionally left blank.

Appendix A: Acronyms and Abbreviations

A.1. Acronyms

BSP	Board Support Package
UML	Unify Modeling Language

A.2. Glossary

International Mobile Telecommunication 2000 (IMT-2000/ITU-2000)

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

EXPloitation

Software layer within Alcatel architecture which provides access to all terminal services / functionalities

Appendix B: References

1 8010.801, "References and Abbreviations (UMTS)", Texas Instruments

2 "System On Chip - LoCosto", Texas Instruments

3 L1S_CALP010, "", Texas Instruments

4 "Triton", Texas Instruments

5 “L1 DRP changes”, Texas Instruments

6 “Location Services”, Texas Instruments

7 “Locosto Use-Case-Analysis, Texas Instruments

8 “Locosto PRD”, Texas Instruments

9 “Locsosto STRQ”, Texas Instruments

10 L1M_SE156_1, “ASD Locosto”, Texas Instruments

11 L1M_GS000, “ASD Locosto”, Texas Instruments

12 “GPF Protocol Stack Framework”, Texas Instruments

13 L1S_LOC_TM026_1, “ASD Locosto”, Texas Instruments

14 “G23M feature set” , Texas Instruments

15 “ARM7 ROM-Code Generation for LoCosto” , Texas Instruments

Appendix C: Annex

C.1.   List of SW components/libraries required for ULC and Locosto+

m	**Mandatory**
n	**not required**
?	**To validate**
o	**Optional**

#N/A means Not applicable

Library name from build	SW Component	Domain	Short description of the SW Component	**Required for ULC**	**Required for L+**

Library	SW	Domain	Description	ULC	Locosto +
abb.lib	ABB	BSP	Analog Base Band driver	m	m
aci.lib	ACI	Modem	PS - Application common Interface	m	m
aci_dti_mng.lib	ACI	Modem	PS - Application common Interface	m	m
aciext.lib	ACI	Modem	PS - Application common Interface	m	m
alr.lib	ALR	Modem	Additional feature of measurement that replace RR	m	m
atiext.lib	ATI	Modem	AT Command Interface	n	?
atp.lib	ATP	Modem	Additional feature of measurement that replace RR	n	m
audio.lib	Audio	Audio & Speech	Audio service	y	*m*
audio_bgd.lib	Audio	Audio & Speech	Audio background	n	*m*
audio_int_ram.lib	Audio	Audio & Speech	Audio background internal RAM	m	*m*
bae.lib	BAE	Audio & Speech	Beatnik Midi engine	m	*m*
bae_sram.lib	BAE	Audio & Speech	Beatnik Midi engine	m	*m*
bmi.lib	BMI	Test & Debug	Basic Mam Machine Interface	m	n
bootloader.lib	Boot	SW Platform	System Boot	m	*m*
bpr.lib	BPR	Audio & Speech		n	*n*
buzzer.lib	Buzzer	BSP	Buzzer	?	?
cama.lib	CAMA	Imaging	Camera Application	n	*m*
camd.lib	CAMD	Imaging	Camera driver	n	*m*
cc.lib	CC	Modem	Call Control - GSM	n	m
ccd_na7_db.lib	CC	Modem	?	n	m
ccddata.lib	CC	Modem	?	n	m
cci.lib	CCI	Modem	?	n	m
cci_ir.lib	CCI	Modem	?	n	m
comlib.lib	COM	Modem	PS utilities	n	m
config_gprs_fl.lib	GPRS	Modem	Config GRPS; Int RAM : 80 k Prog + 118k data	n	m
config_gprs_ir.lib	GPRS	Modem	Config GRPS; Int RAM : 80 k Prog + 118k data	n	m
coreip.lib	IP	Modem	TCP/IP stack	n	m
cry.lib	CRY	Security	Encryption driver	n	n
cst.lib	CST	Modem	Custom AT command	n	m
dar.lib	DAR	BSP	Diagnose and Recovery	m	m
dar_gbl_var.lib	DAR	BSP	Diagnose and Recovery	m	m
dcfg.lib	DCFG	BSP	Data configuration	n	?
dcm.lib	DCM	BSP	Data Configuration Manager	n	?
die.lib	DIE	BSP	Die ID library	n	?
dio_il.lib	DIO	Modem	DIO library	n	m
dl.lib	DL	Modem	Data Link Layer - GSM	m	m
dma.lib	DMA	BSP	Direct Memory Access driver	n	?
dmg.lib	DMG	BSP	DMA task	n	?
drivers_flash.lib	Flash	BSP	Boot ROM	m	m
drivers_int_ram.lib	Flash	BSP	Boot ROM	m	m
dti.lib	DTI	Modem		n	m
etm.lib	ETM	BSP	Enhanced Test Mode: RF settings	m	m
fad.lib	FAD	Modem	Fax Adaptation - GPRS	n	m
ffs.lib	FFS	BSP	NOR Flash File System and driver	m	m
ffsdrv.lib	FFS	BSP	NOR Flash File System and driver	m	m
ffs_drv.lib	FFS	BSP	NOR Flash File System and driver	m	m
ffs_pcm.lib	FFS	BSP	NOR Flash File System and driver	m	m

frame_na7_db_fl.lib	GPF	SW Platform	Generic Purpose Frame	***m***	***m***
frame_na7_db_ir.lib	GPF	SW Platform	Generic Purpose Frame	***m***	***m***
gbi.lib	GBI	BSP	RFS driver	**n**	**m**
gdi.lib	GPRS	Modem	GPRS	**n**	**m**
gil.lib	GIL	BSP	Generic Interface Layer (GPF/RIV)	**n**	**m**
gmm.lib	GMM	Modem	GPRS Mobility Management - GPRS	**n**	**m**
grlc.lib	GRLC	Modem	GPRS Radio Link Control - GPRS	**n**	**m**
grlc_ir.lib	GRLC	Modem	GPRS Radio Link Control - GPRS	**n**	**m**
grr.lib	GRR	Modem	Radio Link in GPRS mode	**n**	**m**
hash.lib	Hash	Security	Hash driver	**n**	*?*
i2c.lib	I2C	BSP	I2C Driver	**n**	**m**
icn.lib	ICN	Applications	Icon of the BMI	**n**	**n**
img.lib	IMG	Imaging	Teleca Codec	**n**	**n**
init_common.lib	OS	SW Platform		***m***	***m***
init.lib	OS	SW Platform		***m***	***m***
ip.lib	IP	Modem	TCP/IP stack	**n**	**m**
jpeg.lib	JPEG	Imaging	JPEG decoder	**m**	**n**
kpd.lib	KPD	BSP	Keypad drvier	**n**	**m**
l1_custom_ext.lib	L1	Modem	Layer 1	**m**	**m**
l1_custom_int.lib	L1	Modem	Layer 1	**m**	**m**
l1_ext.lib	L1	Modem	Layer 1	**m**	**m**
l1_int.lib	L1	Modem	Layer 1	**m**	**m**
l1_intram.obj	L1	Modem	Layer 1	**m**	**m**
l1_pei.lib	L1	Modem	Layer 1	**m**	**m**
l2r.lib	L2R	Modem	Layer 2	**m**	**m**
llc.lib	LLC	BSP	Low Level Charger driver	**n**	**m**
lls.lib	LLS	BSP	Low Level Service driver	**n**	**m**
mc.lib	MMC	BSP	MMC card driver	**n**	**m**
mfw.lib	MFW	Test & Debug	Mobile Framework	***m***	***n***
misc_na7_db_fl.lib	GPF	SW Platform	Generic Purpose Frame	***m***	***m***
misc_na7_db_ir.lib	GPF	SW Platform	Generic Purpose Frame	***m***	***m***
mks.lib	MKS	Security	Magic Key Services	**n**	*?*
mm.lib	MM	Modem	Mobility Management - GSM	**m**	**m**
mpk.lib	MPK	Security		**n**	*?*
nan.lib	NAN	BSP	NAND Driver	**n**	**m**
nucleus_flash.lib	OS	SW Platform	Nucleus OS	***m***	***m***
nucleus_int_ram.lib	OS	SW Platform	Nucleus OS running in internal RAM	***m***	***m***
osx_na7_db.lib	GPF	SW Platform	Generic Purpose Frame	***m***	***m***
pcm.lib	PCM	Modem		**?**	**?**
power.lib	PWR	Power	Power driver	***m***	***m***
ppp.lib	PPP	Modem	PS - PPP server - GPRS	**n**	**m**
ppp_ir.lib	PPP	Modem	PS - PPP server - GPRS	**n**	**m**
psi.lib	PSI	Modem	PSI	**n**	**m**
r2d_custom_flash.lib	R2D	Imaging	Riviera 2D driver	**n**	**n**
r2d_flash.lib	R2D	Imaging	Riviera 2D driver	**n**	**n**
r2d_int_ram.lib	R2D	Imaging	Riviera 2D driver	**n**	**n**
r2d_drv_custom_flash.lib	R2D	Imaging	Riviera 2D driver	**n**	**n**
r2d_drv_flash.lib	R2D	Imaging	Riviera 2D driver	**n**	**n**
r2d_drv_int_ram.lib	R2D	Imaging	Riviera 2D driver	**n**	**n**
ra.lib	RA	Modem	Rate Adaption - GSM	**m**	**m**
rfs.lib	RFS	BSP	Riviera File System driver	**n**	**m**
rfsfat.lib	RFS	BSP	RFS FAT file system	**n**	**m**
rfsnand.lib	RFS	BSP	RFS NAND file system	**n**	**m**
riviera_core_flash.lib	RIV	SW Platform	Riviera frame	***m***	***m***
riviera_cust_flash.lib	RIV	SW Platform	Riviera frame	***m***	***m***

riviera_cust_int_ram.lib	RIV	SW Platform	Riviera frame	*m*	*m*
rlp.lib	RLP	Modem	Radio Link Protocol - GSM	m	m
rnet.lib	RNET	BSP	Riviera Net (TCP/IP)	n	?
rnet_rt.lib	RNET	BSP	Riviera Net (TCP/IP)	n	?
rnet_rt_coreip.lib	RNET	BSP	Riviera Net (TCP/IP)	n	?
rng.lib	RNG	Security	Random Number Generator driver	n	?
rr.lib	RR	Modem	Radio Resources - GSM	m	m
rss.lib	RSS	Security	ROM Secure Service driver: check at boot time	*m*	*m*
rtc.lib	RTC	BSP	Real Timer Clock driver	m	m
rtc_drv.lib	RTC	BSP	Real Timer Clock driver	m	m
rts16le_flash.lib	OS	SW Platform		?	?
rts16le_int_ram.lib	OS	SW Platform		?	?
secure_rom.lib	ROM	Security	Boot ROM	?	?
sim.lib	SIM	BSP	SIM driver	m	m
sim_b_lib.lib	SIM	BSP	SIM driver	m	m
sim_drv.lib	SIM	BSP	SIM driver	m	m
sm.lib	SIMLock	Modem	SIME Lock driver	m	m
sms.lib	SMS	Modem	Short Message Service	m	m
sndcp.lib	SNDCP	Modem	GPRS	n	m
spi.lib	SPI	Connectivity	Serial Port Interface driver	n	m
spi_drv.lib	SPI	Connectivity	Serial Port Interface driver	n	m
ss.lib	SS	Modem	Supplementary Service - GSM	m	m
str2ind.obj	GPF	SW Platform	Generic Purpose Frame	*m*	*m*
t30.lib	T30	Modem		m	m
tif_na7_db_fl.lib	GPF	SW Platform	Generic Purpose Frame	*m*	*m*
tif_na7_db_ir.lib	GPF	SW Platform	Generic Purpose Frame	*m*	*m*
tpudrv.lib	TPU	Modem	Time Processor Unit	m	m
tty.lib	TTY	Audio & Speech		o	*m*
uart.lib	UART	Modem	UART Entity	m	m
uart_b_lib.lib	UART	Connectivity	UART Driver	n	m
uart_drv.lib	UART	Connectivity	UART Driver	n	m
udp.lib	UDP	Modem	UDP protocol	n	m
usb.lib	USB	Connectivity	USB device driver	m	m
usbfax.lib	USBFAX	Connectivity	USB Fax driver	n	m
usbms.lib	USB	Connectivity	USB Mass storage	n	n
usbtrc.lib	USBTRA	Connectivity	USB Trace	m	m
	MDC	Audio & Speech	Melody converter	*n*	*n*
	MDL	Audio & Speech	Melody download	n	n
	RGUI	Applications	Riviera Graphical User Interface service	n	n
	SMBS	Applications	Short Message Bearer service	n	n
	TEAL	Applications	T9 application layer	n	n
	TUT	BSP	Test Unit	n	m
bat.lib	BAT	Modem	Bat library	n	m
mms.lib	MMS	Applications	MMS client	n	n
mmsClient.lib	MMS	Applications	MMS client	n	n
nexgen.lib	TCPIP	SW Platform	TCP/IP stack	?	?
nucleus_flash_nodbg.lib	OS	SW Platform	?	?	?
lcc.lib	LCC	BSP	Low Cost Charger	n	m
main.lib	OS	SW Platform	?	?	?
mic.lib	MIC	Applications	?	n	n
nucleus_int_ram_nodbg.lib	OS	SW Platform	Nucleus	?	?
remu_na7_db.lib	RIVEMU	SW Platform	Riviera Emulator	?	?
socket_lib.lib	TCPIP	Modem	TCP/IP stack	n	m
tcpip.lib	TCPIP	Modem	TCP/IP stack	n	m
wapmic.lib	WAP	Applications	WAP Browser	n	n

	?	m	?
	?	m	?
ıs	ZI Engine	n	n
ıs	ZI Engine	n	n
m	OS interface for L1	?	?
	CCD	m	n
m	GPF Frame	?	?
m	GPF Frame	?	?
	GSM Modem	m	n
	Misc	m	n
	Misc	m	n
	Emuzed JPEG encoding/decoding library	*m*	*m*