Technical Documentation

Migrating from TCS2.1 to TCS3.2

Document Number:	Document number to be assigned
TI Department	European Wireless Terminal Chipset Business Unit
Version:	0.8
Status:	Draft
Date:	March-1-2005

IMPORTANT NOTICE

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

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

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

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

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

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

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

Change History

Date of change	Changed by	Approved by	Date of approval	Version	Notes
1st March 2005	Iain Hunter			0.1	(1)
22nd March	Iain Hunter			0.2	(2)
29th March	Iain Hunter			0.3	(3)
30th March	Iain Hunter			0.4	(4)
5th April	Iain Hunter			0.5	(5)
20th May 2005	Iain Hunter			0.6	(6)
17th June 2005	Iain Hunter			0.7	(7)
21st June 2005	Iain Hunter			0.8	(8)

Notes

(1) Initial version
(2) Incorporated comments from MMI team
(3) Added Tools Issues section. Removed R2D, Added UART trace, NOR only FFS
(4) Added DRP comments and NAND FFS
(5) Added ARM7 memory architecture details and bootload speeds
(6) Added Audio Service and details of Text Entry Abstraction Layer
(7) Removed TEAL. Added CAMIL and IMGIL and MMI driver changes.
(8) Corrected Audio MMI driver volume change

Table of Contents

1 Introduction

The purpose of this document is to summarise the software changes to be made when migrating a customers phone from the TCS2.1 release (Calypso/Iota based) to a TCS3.2 release (Locosto/Triton based).
The document is targeted at both application and MMI engineers and low system software/driver engineers as they have different requirements.

2 TCS3.2 Overview

TCS3.2 is the software release targeted for the Locosto combined Digital BaseBand (DBB) and Digital Radio Processor (DRP) and Triton power management devices.

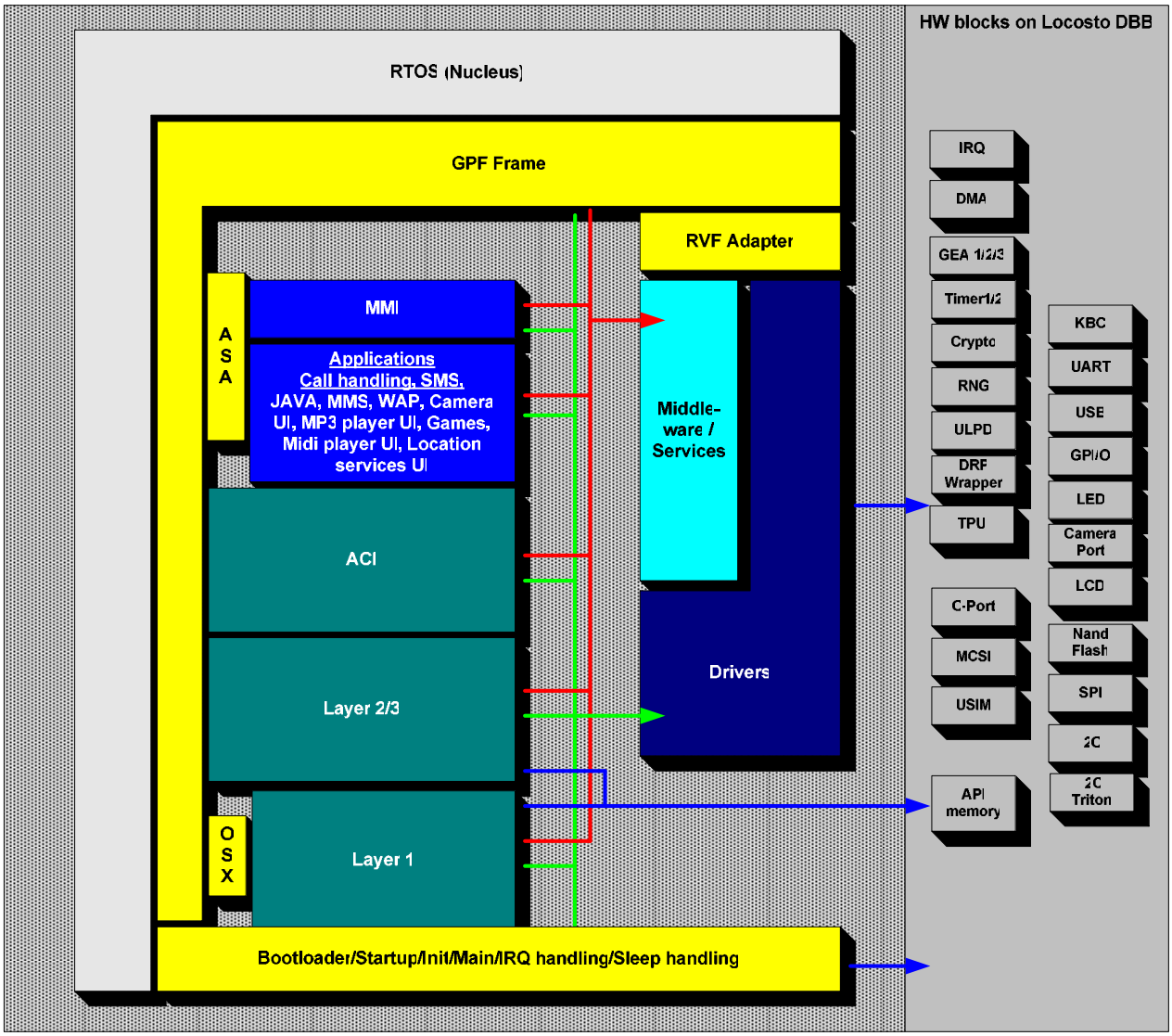

Figure 1:TCS3.2 SW Architecture on ARM7

The boot-loader allows programming the flash via a USB connection. The boot loader runs from a special secure ROM area of the chip and which uses the crypto capabilities of LoCosto to ensure that only certified software downloads can be performed.
Startup/Init is the code which is performed after reset to configure the hardware and startup software tasks. It is responsible for the following initialisations:

- Clock and memory timings
- Hardware peripherals.
- Static variables from cinit segment and zeroing of memory.
- SW drivers, services and the RTOS so that the scheduling of threads can be started when the system starts up.
- Starting the DSP, loading the DSP patch file and DRP initialization script.

The IRQ handling is responsible for mapping hardware interrupts into calls into the interrupt driven drivers and the synchronous L1.
Sleep handling is tightly coupled with the L1 Synchronous (L1S) processing. It manages the different sleep modes, makes the sleep decisions and performs the time correction after wake-up.

Drivers are responsible for providing a low level API to access the functionality of each peripheral hardware block.
Middleware/Services provide more complex functionality to the MMI and applications using the underlying drivers. Examples of services/middleware are:

- DAR Diagnose and Recovery
- JPEG encoding/decoding
- Flash File System (FFS) on NOR and NANDFlash
- A camera system which controls a camera sensor via two different HW interfaces, and manage the programming of the DMA between the sensor, external memory and the LCD in order to support camera preview and snapshot.
- MIDI player engine
- MP3 Player

The Real Time Operating System (RTOS) used is the same version of Nucleus as in TCS2.1.
In order to be agnostic to the RTOS several adaptation layers are need for different parts of the system.
The GPF frame is used by the L23, Application Command Interface (ACI) and MMI to create threads, allocate memory, send primitives, start/stop timers etc.
L1 uses a mixture of calls to an abstraction layer (OSX) and direct Nucleus calls to get access to the system resources. OSX is a glue layer which maps the L1 system interface requirements to the GPF API.
Drivers, Services and the trace part of L1 use RiViera Frame (RVF) abstraction layer to create threads, allocate memory, send primitives, start/stop timers etc.

In TCS2.1 the drivers and services and some parts of L1 use the Riviera frame (RVF) API to access the operating system. This meant that there were two frameworks, Riviera and GPF, embedded in the system. As these two frameworks are both based on Nucleus, to reduce the memory footprint TCS3.2 will replace the RVF framework with a thin glue layer which maps the RVF API to the GPF functionality. With this approach the driver and service code does not need to be changed to a new interface

The fact that a 3rd party Application suite and MMI are also targeted in TCS3.2 requires another layer to access the RTOS. Since the requirement of the application suite is not fully clear today this glue layer gets the temporary name Application Suite Adaptation Layer (ASA).
L1, L23 and ACI represent the modem software of the phone. GSM only or GSM voice and CSD + GPRS packet switched operations can be provided. On top of ACI an AT-Command based interface is provided to enable application to control the modem.
Applications are seen as separate functional blocks which manage several areas of the phone's UI. Typical examples are browsers for WAP, Editors for SMS and MMS, Personal Information Manager, Phonebook manager, Games, Camera UI and Idle screen. Applications are controlled by a MMI which controls the activation, suspension and deactivation of the applications. It also arbitrates between applications for access to the human interface I/O blocks keypad, microphone, loudspeaker and display.

3 Tools Migration Issues

There are several changes in the tool chain required when moving to TCS3.2.

3.1 Compiler Chain

TCS3.2 will move to a version 2.54 of the TI ARM code generation tools. This is a C/C++ compiler. It will replace the v1.22 compiler and v1.9902 Visual Linker used in TCS2.1.

3.2 Build System

TCS3.2 will use the same Unified BusyB build system as TCS2.1. There will of course be a different linker command file template to reflect the different memory map of LoCosto and the fact that some code is ROMed.

3.3 Bootloading

LoCosto must boot via the internal secure ROM bootloader and so the external boot option from TCS2.1 is lost. The secure ROM bootloader has the following options to download a binary image

- USB option added at 12Mbps
- UART modem at 3.25Mbps

The IrDA option is no longer supported for bootloading.

The binary image must contain a security certificate to authenticate the image. This certificate will be automatically created within the compile tool chain.

3.4 Cellular Systems Software Tools (CSST)

LoCosto will use a new PC tool called CSST to manage RF calibration, Flash programming, image signing and FFS formatting. This tool will replace the TMSH and FLUID tools used by TCS2.1 whilst adding the security features.

3.5 Emulator Debugging

By default LoCosto has emulator access disabled. It can only be enabled by the secure ROM boot-loader if a flag is set in the security certificate. It will be possible to debug a production image (with emulation disabled) by presenting the bootloader with a new security certificate via USB or UART at reset which will enable emulator access.

4 Software Migration Issues

The major issue to be dealt with when migrating from the Calypso-Lite DBB to LoCosto DBB is handling the modified or new hardware peripherals. There are four major blocks in the software which are each affected differently.

- MMI and Applications
  The MMI and application should not directly touch the hardware (hw). They should always go through the Services and Drivers. In most cases these APIs are unchanged or extended to support new functionality so there should be minimal impact.
  The TI BMI will make the following changes:
  - Audio driver (g23m\condat\com\src\driver\audio.c) will now control the volume via audio service API (chipsetsw\services\audio\audio_api.c) rather than directly via ABB driver.
  - Backlight control will be via the LLS service (chipsetsw\services\lls\lls_api.c) rather than Buzzer driver (chipsetsw\drivers\drv_app\buzzer\buzzer.c)

- Secure IMEI access will be via sAT_Dn() rather than cl_get_imeisv().

- Services and Drivers
  The primary aim of these drivers is to abstract away from the applications any change in underlying functionality caused by the change in DBB. In most cases there is an unchanged API for the previous level of functionality. In many cases the LoCosto peripherals provide new or enhanced functionality which will lead to a new or extended API. This should not have a significant impact on migrating existing MMI and Applications as they will be able to use the previous APIs as is.

- Modem (Application Command Interface and Layers 1/2/3)
  As supplied by Texas Instruments these modules will already have compile flags to deal with any changed peripherals. Therefore, there should be no impact to migration.

- System Startup
  The low level software that manages system startup will be the most impacted by the migration as it will need to be rewritten to ensure that all available peripherals are correctly initialized and started.

Each of these blocks will now be dealt with in more detail in sections 4.1 to 4.4

There are significant changes in ARM7 architecture moving from Calypso to LoCosto.

- ARM7 now runs at 104MHz.
- There is 2.5MBits of 32 bit wide internal SRAM accessible at 104MHz.
- There is 1.5MBits of 32 bit wide internal ROM code containing critical MIDI and Image codec code accessible at 104MHz.
- The external bus has a prefetch buffer of 4x32bits to speed up linear accesses to burst mode Flash memories.

4.1 MMI and Application Migration

The MMI and applications should be abstracted from the changes in the chipset by the services and drivers layers and so should not need to be changed to achieve the same level of functionality. There are two modules from TCS2.1 that will be removed in TCS3.2 as they are historic Riviera drivers that are no longer used.

- The Agnostic Transport Protocol (ATP) service. This service allows an application to open a channel to the modem (ACI) in TCS2.1. Although ATP is part of a TCS2.1 delivery, it is not used in the standard implementation and so its removal should not cause any issues. In TCS3.2 ATP is replaced by the Soft multi-Channel Shared Memory Interface (S-CSMI).
- The Riviera 2D (R2D) driver. In a GPF based MMI the graphics are controlled by the file display.c and it makes only a few calls into R2D. In TCS3.2 R2D will be replaced by a much smaller low level LCD driver that will be called by display.c.

One other area that may need investigation is the setting of audio modes (eg handset or handsfree) and settings such as volume. Assuming that these parameters are controlled via configuration files (ie use of audio_mode_load() API) in the FFS the MMI need not be aware of the underlying changes in file contents due to the different audio paths and DSP audio filtering options. The contents of these files will need to be changed to the TCS3.2 format but this will be hidden from the MMI code.

4.2 Services and Driver Migration

The services and drivers fall into four categories in the migration to TCS3.2 as shown in Figure 2. They are:

- API and implementation unchanged (Type 1)

There is no impact with these modules.

- API unchanged but implementation changed (Type 2)
  There is no impact to the application/MMI as the API is unchanged. Most of the services and drivers in this category control Triton. This is because the underlying communication mechanism with the power management chip has moved from SPI in TCS2.1 to I2C in TCS3.2 for Triton.

- API extended and the implementation changed (Type 3)
  The Audio service is in this category as chipset supports extra functionality in TCS3.2 such as DSP audio signal processing, MIDI Player, DSP MP3 playback and the additional stereo audio DAC and audio paths in Triton.
  With this module the new APIs and parameter definitions are implemented as extensions to TCS2.1 implementation and so existing driver usage should continue to work.

- New Module (Type 4)
  These are modules providing entirely new functionality to TCS3.2 using the enhanced features of the Locosto/Triton chipset. In the case of the S-CSMI, USIM and LCD drivers these are replacements for the ATP, SIM and R2D drivers in TCS2.1.

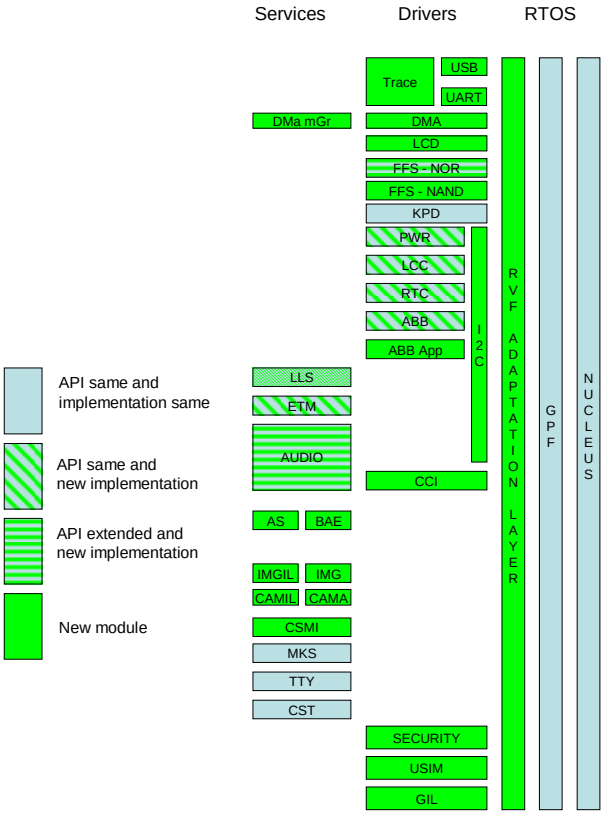

Figure 2: Services and drivers

Each of the services and drivers will now be summarised and categorised.

4.2.1 Services

DMAG (Type 4)
DMA manaGer service controls access to the 6 DMA channel and arbitrates between requesting applications and services for use of DMA channels.

DAR (Type1)
Diagnosis and Recovery module is unchanged

LLS (Type 2)
Low Level Services module controls the 3 LED drivers in Triton. Its implementation has changed as it now uses I2C to communicate with Triton

ETM (Type 2)
Enhanced Trace Module communicates with a PC tool (TestMode SHell) via USB driver to allow calibration and configuration of the phone. Its API remains the same as it is used purely to start and register ETM module in the system. Its implementation will be enhanced to support the new Audio features and use of I2C to communicate with Triton.

AUDIO (Type 3)
Audio services control the DSP audio features, Triton audio paths and audio configurations. The API has been extended in TCS3.2 to support all the new features such as MP3 and the new audio routing options supported by Triton. It uses I2C drivers to configure Triton and Stereo C-Port driver (CCI) to configure I2S interface for audio to Triton.

AS (Type 4)
Audio Service is responsible for managing the use of the MIDI player (BAE service) either as a MIDI ringer or a simple MIDI player. It is only service that should access BAE.

BAE (Type 4)
Beatnik Audio Engine is used to access the Beatnik MIDI services. Time critical parts of this service and a 100k DownLoadableSound (DLS) library are ROMed in the ARM7 memory on Locosto. The BAE module should not be accessed directly by the MMI. The AS should be called to access and manage MIDI functionality

IMGIL (Type 4)
IMaGe Interface Layer used by MMI to access Image compression module.

IMG (Type 4)
IMaGe compression module is ROMed in LoCosto and allows JPEG encoding and decoding of BMP, GIF, wBMP, PNG and RAW_YUYV formats.

Camera Interface Layer (Type 4)
Interface layer used by MMI to access the Camera service.

Camera (Type 4)
Camera service is responsible for controlling the Camera sensor, managing the data transfers with DMA and image format transforms using IMG module

S-CSMI (Type 4)
The Soft multi Channel Shared Memory Interface is used to provide multiple data channels between applications and the modem. It replaces the ATP module in TCS2.1.

MKS (Type 1)
Magic Key Sequence module allows key sequences to be registered and then intercepted from the Keypad driver.

TTY (Type 1)
Touch TelephonY services allows audio profile for the CTM modem to be configured

CST (Type 1)
Customer Specific Task (CST) is a generic task that allows addition of new customer specific AT commands.

4.2.2 Drivers

Trace (Type 4)
The USB trace driver allows sending and receiving of traces and primitives from GPF frame. It is DIO compliant. This replaces the Riviera Trace mechanism in TCS2.1. It uses either USB or UART as transport mechanism.

USB driver (Type 4)
Manages the USB transceiver. It is not a USB stack.

UART driver (Type 4)
New UART driver called by trace driver

DMA (Type 4)
Provides API access to control registers to initialise and use DMA channels. It will only be accessed by DMA manager and not directly by applications.

LCD (Type 4)
Provides low level functions to control the LCD. The MMI will access these via display.c

FFS – NOR (Type 3)
This is the NOR flash FFS driver from TCS2.1. It has been extended to allow “files” to exist as linear arrays in memory.

FFS- NAND (Type 4)
Provides a file system using the NAND Flash.

I2C (type 4)
The new I2C driver allows a single API to be used to control the 2 hardware I2C modules. One interface is for Triton and the other is available for external devices.

LLC (Type 2)
Low Cost Charger module controls the charging process in Triton. Its implementation has changed as it now uses I2C to communicate with Triton.

RTC (Type 2)
Real Time Clock is now in Triton and so driver uses I2C to access it.

ABB (Type 2)
Analogue BaseBand driver now uses I2C to access Triton registers.

ABB_App (Type 4)
Analogue BaseBand driver for applications.

CCI (Type 4)
Codec Control Interface (CCI) driver is used to configure the C-Port (Codec Port) I2S audio interface to Triton.

Security (Type 4)
This driver allows access to the secure services in the ARM7 such as Random Number Generator (RNG), Cryptography module (DES/3DES) and hashing module (MD5/SHA-1).

USIM (Type 4)
The Smart Card SIM driver replaces the old SIM driver in TCS2.1.

GIL (Type 4)
Gpf Interface Layer acts as interface between Riviera Services and Drivers and MMI. It allows Riviera callback functions to send a GPF message to the MMI

4.3 Core Modem Functionality

The core modem functionality supplied by L1, L2/3 and ACI is essentially unaffected by the transition from TCS2.1 to TCS3.2. Both of these releases are generated from TI from the same baseline release and so have the extended functionality of TCS3.2 enabled by compile flags.
The peripherals touched by these modules are:

- Application Command Interface
  HW security modules (Crypto, RNG) and USIM to provide SIM lock and IMEI protection

- Layer 2/3
  The Link Layer Controller (LLC) layer directly accesses the GEA module which now incorporates GEA3 algorithm.

- Layer 1
  The Digital Radio Processor (DRP) is controlled by the Layer 1 using the same TPU interface as in Calypso so there is no fundamental impact from the integrated RF. TI will supply a new set of TPU drivers to control DRP which will be included in the build via compile flags just like any other RF.
  The embedded DSP will be accessed via the Arm Processor Interface as in Calypso so again this will be handled by compile Flags.

4.4 System Startup

The system startup routines will need to be significantly changed to configure LoCosto device for operation. The drivers involved will be the MEMory InterFace (MEMIF), CLock Management (CLKM), DPLL, Interrupt handling, RHEA peripherals and ARM IO driver.

In addition to downloading the patch file to the DSP at initialization, the ARM must also download the DRP script processor configuration file. This file contains:
- the basic initialisation of DRP script processor
- the translation of TPU commands into DRP command sequences