**CONTENTS PAGE**

1 Abbreviations 4

2 References 4

3 General 5

4 Hardware Interfaces 5

5 Software Architecture 6

5.1 General 6

5.2 Startup 7

5.3 SYS/BIOS 7

6 SW processes and interfaces 9

6.1 Appl\_A 9

6.2 Appl\_D 9

6.3 RN 9

6.4 RN\_PHY 9

6.5 TrueVoice 9

6.6 Recordings 9

6.7 Maintenance 9

6.8 Common Interfaces 10

7 CATEL-1451, Base Station Application 13

7.1 Startup and Initiation 13

7.2 Common processes 13

7.3 Appl\_A 13

7.4 Appl\_D 14

8 CATEL-1452, Personal Transceiver Application 15

8.1 Startup and Initiation 15

8.2 Common processes 15

8.3 Appl\_A 15

8.4 Appl\_D 16

9 CATEL-1454, Communication Adapter Application 17

9.1 Startup and Initiation 17

9.2 Common processes 17

9.3 Appl\_A 17

9.4 Appl\_D 18

**Document History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Issue** | **Issue Date** | **Section** | **Description of change** |
| 0.1 | 2013-06-14 |  | First Draft |
| 0.2 | 2013-06-19 |  | Continued |

# Abbreviations

BC Battery Charger

BS Base Station

CA Communications Adapter

CLT Command Line for Take of Command

FTN/ATL Defense Telephony Network

PT Personal Transceiver

VOX Voice Operated Exchange

EEProm Electrically Erasable Programmable Read-Only Memory

LED Light Emitting Diode

MCU Micro Controller Unit

PTT Push-to-Talk

RTOS Real-Time Operating System

TBD To Be Defined

TDM Time Division Multiplexing

UART Universal Asynchronous Receiver Transmitter

# References

1. 12FMV495-41:3 Technical Specification
2. 1450-SYSPS-001 System and Products Specification
3. 1450-SDD-002 Radio Network Description
4. 1450-SDD-003 Radio Network Software API
5. 1453-SDD-001 Specification Battery Charger
6. RTCA/DO-178B Software considerations in airborne systems and equipment  
    certification
7. 1450-HSID-001 CPU pin allocation
8. SYS/BIOS Real-Time Operating System

# General

The Software Design Description defines the low level requirements on the Software for the CATEL-1450 LUF System. The low level requirements are developed from the 1450-SYSPS-001 [2] document that defines the high level system function requirements. The low level requirements are expressed in text and not in uniquely identified requirements. The reason for this is that the software is developed to software safety level D, according to RTCA/DO-178B [6], that defines that the low level requirement do not need to be explicit verified but verified through the verification of the high level requirements.

Following CATEL 1450 LUF System units contain software described in this document:

* CATEL-1451, Base Station (BS)
* CATEL-1452, Personal Transceiver (PT)
* CATEL-1454, Communication Adapter (CA)

All units have their own Executable Object Code loaded which is built by common routines and product specific routines. Some functional parts are sourced as lib, see ref[x]-[x]

The software in the CATEL-1453 Battery Charger (BC) has a separate HW and SW architecture and is excluded from the common software functions here described. See 1453-SDD-001 [5].

# Hardware Interfaces

The hardware processor architecture is in larger parts common to all units. They consist of an OMAP-L138 C6-Integra™ DSP+ARM® Processor from Texas Instruments. The device consist of an ARM926EJ-S 32-bit RISC processor core and a C674x DSP core. Example of peripheral sets that are used of the system units are: inter-integrated circuit (I2C) Bus interfaces; multichannel audio serial port (McASP); SPI interfaces with multiple chip selects; 64-bit general-purpose timers each configurable (one configurable as watchdog); general-purpose input/output (GPIO) with programmable interrupt/event generation modes; UART interfaces; external memory interfaces.

Analogue inputs and outputs for audio are performed with external, to the MCU, audio codecs with their TDM data bus connected to the McASP port.

Switches and discrete inputs and outputs are connected to the MCU general purpose IO pins.

The hardware interface to the MCU describing all physical connections can be found in the 1450-HSID-001 [7] document.

# Software Architecture

## General

The software is divided into two separate sections. One part is executing in the ARM core, handling the logical functionality of the unit. The other section is executing in the DSP core, handling the real time audio processing and mixing. Both ARM and DSP functions are implemented as separate tasks scheduled by a multitasking pre-emptive real time operating system, SYS/BIOS [8], and various device drivers, both sourced and own developed. The two sections interact through IPC routine.

Headphone  
DAC

TrueVoice

I2S

Headphone  
MIC \* 2

Codec

SPI

Appl\_D

Appl\_A

RN

Radio

SPI

RN\_PHY

DIO

DIO

LEDs

Keyboard

Key pressed

LED  
Ctrl

Config

Audiodata

Payload

Config

Appl data DSP<->ARM

DSP

ARM

Payload

Config

Payload

TRACE

TRACE

Parameters

Main-  
tenance

UART0

Flash

SPI

Rec Audio

Light sens

I2C

Light Sens

I2C

Headset ctrl

Power sup

Power  
Supp

CLI

UART2

IRQ

Radio data/ctrl

Recordings

SYS/BIOS

SYS/BIOS

Radio  
pwr

Headset  
ctrl

Hardware

Application SW

Mediate SW  
Spec algorithms

Low level SW interface

Support SW, OS

Flash

Calls

Data flow/payload

## Startup

The first part of the main routine performs the unit common P-BIT functions that

* test the check sum of the program
* test the codec connection
* test the EEProm function

The main routine starts then IPC and SYS/BIOS and synchronize the startup with the DSP by the IPC attach function. The SYS/BIOS starts to schedule the static processes. Any start orders for the processes are not defined. Every process must, in given order:

* Initiate hardware they going to use. The unit initiation is described per unit later,
* Create queues for incoming messages (SYS/BIOS)
* Wait for creation of external message queues (SYS/BIOS)

### Watchdog

A watchdog, preferable hardware watchdog, is used to make sure the software not hanging forever.

A low prioritized process, not the idle process, shall be kicking the watchdog every second and the watchdog shall restart the software if watchdog not kicked within five seconds.

## SYS/BIOS

The operative system, RTOS, running in both the ARM and in the DSP core is SYS/BIOS. The OS are complemented with IPC and parts of Starterware.

IPC is an additional package to SYS/BIOS to handle messages within a CPU as well as between cores.

Starterware contain driver interfaces like uart, dma, spi etc. Starterware are written for a non-os system and can interfere with SYS/BIOS.

SYS/BIOS and additional packages are provided by Texas Instrument.

### Tick

A system tick is set to 10 ms. a separate timer is used by PHY for a more accurate timing.

### Message queues (SYS/BIOS/ICP)

Inter Process Communication (IPC) component is used for signaling between tasks. IPC are an optional component to the operating system SYS/BIOS.

Servers creating a message queues with global defined name. A server is process (task) expecting a requests from other processes. The clients are using a name server to find servers. The reply address in message header must be set by the client.

A message can only have one or one of a few different sizes. This limitation is to avid memory fragmentation. The software requests a size of a message buffer and the system returns a buffer of a proper size. The different size or sizes are defined by the configuration.

Example of using the messages:

MessageQ\_create

MessageQ\_open(server)

MessageQ\_create

Server not found

Server found

MessageQ\_alloc

MessageQ\_put

MessageQ\_put

MessageQ\_delete

Client

Server

Create message queue

Wait for server queue to be created

Create message queue

Alloc request signal

Send request to the server

Server reuse message buffer, reply to sender

Client free the signal

SYS/BIOS/IPC

# SW processes and interfaces

## Appl\_A

The ARM application is specific for each system unit and is described in the per unit description chapters.

The application is running in a SYS/BIOS environment and communicates with DSP application over shared memory handled by MessageQ (IPC).

## Appl\_D

The DSP application is specific for each system unit and is described in the per unit description chapters.

The application is running in a SYS/BIOS environment and communicates with ARM application over shared memory handled by MessageQ (IPC).

## RN

Handle the Logical Radio network. Described in document 1450-SDD-002 [3]

## RN\_PHY

Handle the Physical Radio network. Described in document 1450-SDD-002 [3]

## TrueVoice

TrueVoice handle all audio and sound processing such as filters, squelch and noise gates/reduction, mixer functions etc.

Described in document 1450-XXX-XXX [x]

## Recordings

Get sound files from a storage. The process has functions to select an audio file, start and stop the audio playing. The actual audio sample for the sound is managed by the Appl\_D and TrueVoice routine.

## Maintenance

The maintenance process is the motor for the CLI.

For production and maintenance purposes it registers commands like:

* Read/set serial number (only once)
* Read/set date (only once)
* Clear stored data
* update software
* read logs
* etc.

### CLI

CLI is a Command line interface for test and debug. A terminal can be connected to the system units and communicate with processes by commands. This interface is protected for the end user.

Any process can register a command to the interface.

## Common Interfaces

### DIO

Digtal In–Out.

This interface handles buttons, LEDs and radio interrupts.

#### Buttons

Button reading is timer or interrupt driven and need to filter button glitches. To decode what buttons are pressed the driver needs to scan the buttons row and columns.

Clients can subscribe for button events.  
The button function is: subscribe(buttons, event, callback)

where

buttons is a bit pattern representing what buttons to subscribe for.

events can be press/release, click or hold/release.

callback is the function to be called when requested events occur.

#### LED

The LEDs are connect directly to the digital io:s on the cpu.

A client can set the state of the LEDs.

The LED function is: SetLED (leds, color, state)

where  
leds is a bit pattern representing what LEDs to control.

color is the color of the LED. (red, green, yellow).

state is the new state for the LED (off, on).

LEDs intensity by adjust if PWM shall be managed here. Additional function parameter.

#### Display

#### Radio Interrupts

Hardware interrupts from the radio are connected to digital inputs.  
A client can subscribe on interrupts and a callback function is activated when an interrupt occur. The callback will be running in an interrupt context.

The subscribe function is: subscribeIO(io, callback(io))  
where  
io is a predefined interrupt

callback is the function to be called when a interrupt occur.

### Flash

Interface to a storage area, flash

### Headset ctrl

Interface for the headset control. Incl. initial configuration of Codec. Digital pot.

### I2C

The low level interface for the I2C bus. This interface shall only be used by other interfaces.

### I2S

This is the interface for the I2S bus.

### Light sens

This is the interface for the light sensor.

### Param

Interface to store and read permanent parameters. Parameters can be filter parameters or production data. Data is stored into a flash memory. Parameters are handled by a parameter process. Parameters can be set of end users such as: VOX/PTT.

### Power Supp

Interface for the power supply. All units that include battery supply shall have battery alarm warnings.

### SPI

This is the interface for the SPI bus and is used to exchange data with other SPI-units.

SPI is DMA driven. The clients must call the initialization function before using the SPI for the first time.

Function: Init(bus,force).

This is the initialization of the bus where:

bus is what bus to initialize, Radio, codec etc

force flag will do an initialization even if it’s already done.

Function: exchange (bytes, outbuffer, inbuffer, callback)

This function sends and receives data.

Parameter bytes are the number of bytes to send/receive.

Parameter outbuffer is a pointer to data to send. It can be a null pointer.

Parameter inbuffer is a pointer to data area to save incoming data. It can be a null pointer and can be the same as outbuffer.

Parameter callback is the function to be called when all data is transferred. The callback is running in the context of an interrupt.

### Timer

A high resolution timer is used by the PHY to synchronize radio time slots. Timer function is included in SYS/BIOS.

### Trace

Implements a trace interface and can be used by any software. A Trace is a printable string and it can be stored in memory or directed to a serial output.

# CATEL-1451, Base Station Application

The Base Station is a stationary unit that mainly facilitates the communication paths between the remote command center and the aircraft pilot. The Base Station also provides the communication between technicians and the Defense Telephony network as well as the pilot communication with the different technicians and finally the communication between the Land Mobile Radio and the technicians or the pilot. The Base Station Application is responsible to provide the LUF System with an:

* interface to the CLT & FTN/CLT
* interface to the FTN/ATL
* interface to the LMR
* interface to the Aircraft
* interface to two Headsets

## Startup and Initiation

At startup the software executes the common startup functions described in 5.2.

The initiation function initiates:

* the codec
* the switches and pushbutton data buffers
* the common memory buffers

## Common processes

Apart from the unit specific Appl\_A and Appl\_D, following common processes, from chapter 6.8, are included to form the Base Station unit:

* RN
* RN\_PHY
* TrueVoice
* Recordings
* Maintenance

## Appl\_A

This is the part of the unit application that is executed in the ARM processor core. The requirements are implemented/realized in different OS tasks.

* the radioTask with the radio network management
* the telephoneTask with line communication
* the audioTask switching rules and interface to Appl\_D
* the mmiTask handle push/keypad buttons and display

### radioTask

The interface to the RN module is handled by the radioTask. Performing RN module required start, setup and re-configuration.

### telephoneTask

The interface to the FTN/ATL and FTN/CLT connection is handled by the telephoneTask. The functionality is implemented as state machine with the states: Hook-On (Idle), Ring, Call, and Hook-Off.

### audioTask

Interface to the Appl\_D and further to the audio processing module TrueVoice. This task collects switching information from management channel and manages which sound shall be sent to which audio output.

### mmiTask

The interface to the user by push/keypad buttons and display is handled by the mmiTask. Operators command thru pushbutton or sequences of pushes are decoded and sent to corresponding task. The guidance thru spoken message menu system is handled by this task.

Pushbuttons include:

* the manual PTT activation
* keypad, TBD

LEDs include:

* display, TBD

## Appl\_D

This is the part of the unit application that is executed in the DSP processor core.

* the application task that manages:
  + the TrueVoice operation
  + the headset Audio communication

# CATEL-1452, Personal Transceiver Application

The Personal Transceiver is a hand held portable unit that connects technicians to the Base Station in a LUF system. The Personal Transceiver Application is responsible to provide the technicians with an:

* interface to Base Stations services
* interface to selectable functionality menus
* interface to own Headsets

## Startup and Initiation

At startup the software executes the common startup functions described in 5.2.

The initiation function initiates:

* the codec
* the switches and pushbutton data buffers
* the common memory buffers

## Common processes

Apart from the unit specific Appl\_A and Appl\_D, following common processes, from chapter 6.8, are included to form the Base Station unit:

* RN
* RN\_PHY
* TrueVoice
* Recordings
* Maintenance

## Appl\_A

This is the part of the unit application that is executed in the ARM processor core. The requirements are implemented/realized in different OS tasks.

* the radioTask with the radio network management
* the audioTask switching rules and interface to Appl\_D
* the mmiTask handle pushbuttons and leds

### radioTask

The interface to the RN module is handled by the radioTask. Performing RN module required start, setup and re-configuration.

### audioTask

Interface to the Appl\_D and further to the audio processing module TrueVoice. This task collects switching information from management channel and manages which sound shall be sent to headphone audio output.

### mmiTask

The interface to the user by push/keypad buttons and LEDs is handled by the mmiTask. Operators command thru pushbutton or sequences of pushes are decoded and sent to corresponding task. The guidance thru spoken message menu system is handled by this task.

Pushbuttons include:

* the manual PTT activation
* Volume +
* Volume –
* Group select
* Tx Select
* Menu Select (5)

LEDs include:

* On-line / Off-line, Green/Red (1)
* Tx destignation select / Group select, Green/Red (4)

## Appl\_D

This is the part of the unit application that is executed in the DSP processor core.

* the application task that manages:
  + the TrueVoice operation
  + the headset Audio communication

# CATEL-1454, Communication Adapter Application

The Communications Adapter is a hand-held device that enables voice communication between the Pilot in the Aircraft with the turn-around team Technicians in situations where a Base Station is not present or needed. The connection to the aircraft is by wire. The device may have a Technician headset attached directly by wire or wireless connection through the radio network. The Communication Adapter Application is responsible to provide the LUF System with an:

* interface to the Aircraft
* interface to two Headsets
* interface to Technicians

## Startup and Initiation

At startup the software executes the common startup functions described in 5.2.

The initiation function initiates:

* the codec
* the switches and pushbutton data buffers
* the common memory buffers

## Common processes

Apart from the unit specific Appl\_A and Appl\_D, following common processes, from chapter 6.8, are included to form the Base Station unit:

* RN
* RN\_PHY
* TrueVoice
* Recordings
* Maintenance

## Appl\_A

This is the part of the unit application that is executed in the ARM processor core. The requirements are implemented/realized in different OS tasks.

* the radioTask with the radio network management
* the audioTask switching rules and interface to Appl\_D
* the mmiTask handle pushbuttons and leds

### radioTask

The interface to the RN module is handled by the radioTask. Performing RN module required start, setup and re-configuration.

### audioTask

Interface to the Appl\_D and further to the audio processing module TrueVoice. This task collects switching information from management channel and manages which sound shall be sent to Technician resp. Pilot audio output.

### mmiTask

The interface to the user by push/keypad buttons and LEDs is handled by the mmiTask. Operators command thru pushbutton or sequences of pushes are decoded and sent to corresponding task. The guidance thru spoken message menu system is handled by this task.

Pushbuttons include:

* the manual PTT activation
* Technician Volume +
* Technician Volume –
* Pilot Volume +
* Pilot Volume –
* Menu Select (5)

LEDs include:

* On-line / Off-line, Green/Red (1)

## Appl\_D

This is the part of the unit application that is executed in the DSP processor core.

* the application task that manages:
  + the TrueVoice operation
  + the headset Audio communication

# 