

|  |  |
| --- | --- |
|  | **imotion** |
|  |  |
|  | TFT 2.0 System |
|  |  |
|  | Complete System Overview |
|  |  |
|  | Technical Description |
|  |  |
|  | Version: 3.0 |
|  | State: Released |
|  | Classification: Internal use only |
|  | Author: RAN |
|  | Creation date: 2013-10-09 |
|  | Repository: $/Gorba/Main/Motion/SystemManager/Documents/TD\_TFT 2.0 System |
|  | Gorba AG  Sandackerstrasse  9245 Oberbüren  Switzerland |

**Table of contents**

[1 Introduction 5](#_Toc391629669)

[1.1 Scope 5](#_Toc391629670)

[1.2 Intended Audience 5](#_Toc391629671)

[2 Product Overview 5](#_Toc391629672)

[2.1 System Overview 5](#_Toc391629673)

[2.2 Different Hardware modes 5](#_Toc391629674)

[2.2.1 Master-Only mode 5](#_Toc391629675)

[2.2.2 Master-Slave mode 6](#_Toc391629676)

[2.2.3 Headless Master-Slave mode 6](#_Toc391629677)

[2.2.4 Master unit 7](#_Toc391629678)

[2.2.5 Slave unit 7](#_Toc391629679)

[2.3 Hardware of TFT 2.x system 7](#_Toc391629680)

[2.3.1 InfoVision Compact, Quadro, Topbox PC-2 and Topbox Mini 7](#_Toc391629681)

[2.3.2 Inform Compact and Topbox 7](#_Toc391629682)

[2.4 Folder structure on the Topboxes 7](#_Toc391629683)

[2.4.1 Config folder 8](#_Toc391629684)

[2.4.2 Data folder 8](#_Toc391629685)

[2.4.3 Log folder 8](#_Toc391629686)

[2.4.4 Presentation folder 8](#_Toc391629687)

[2.4.5 Progs folder 8](#_Toc391629688)

[2.5 Components 9](#_Toc391629689)

[2.5.1 System Manager 9](#_Toc391629690)

[2.5.2 Update 9](#_Toc391629691)

[2.5.3 Hardware Manager 9](#_Toc391629692)

[2.5.4 Protran (Protocol Translator) 10](#_Toc391629693)

[2.5.5 Composer 10](#_Toc391629694)

[2.5.6 DirectX Renderer 11](#_Toc391629695)

[2.5.7 Audio Renderer 11](#_Toc391629696)

[2.5.8 AHDLC Renderer 11](#_Toc391629697)

[2.6 Connection between software components 12](#_Toc391629698)

[2.6.1 System Manager ⬄ Update 12](#_Toc391629699)

[2.6.2 System Manager ⬄ Protran, Composer, DirectX Renderer, Audio Renderer, AHDLC Renderer, Hardware Manager and Update 12](#_Toc391629700)

[2.6.3 Protran ⬄ Hardware Manager 12](#_Toc391629701)

[2.6.4 Protran ⬄ Composer 12](#_Toc391629702)

[2.6.5 Composer ⬄ Renderers 12](#_Toc391629703)

[3 Hardware Information 13](#_Toc391629704)

[3.1 Inputs and Outputs 13](#_Toc391629705)

[3.1.1 InfoVision Topbox PC-2 platform 13](#_Toc391629706)

[3.1.2 InfoVision Compact, Quadro and Topbox Mini platforms 13](#_Toc391629707)

[3.1.3 Atmel controller on InfoVision TFT hardware 13](#_Toc391629708)

[3.1.4 Gorba Inform Topboxes and Compacts 13](#_Toc391629709)

[3.2 Software components interaction with Hardware 14](#_Toc391629710)

[3.2.1 System Manager interaction with hardware 14](#_Toc391629711)

[3.2.2 Update interaction with hardware 14](#_Toc391629712)

[3.2.3 Protran interaction with hardware 14](#_Toc391629713)

[3.2.4 DirectX Renderer interaction with hardware 14](#_Toc391629714)

[3.2.5 Audio Renderer interaction with hardware 14](#_Toc391629715)

[4 System Management 15](#_Toc391629716)

[4.1 System Manager configuration 15](#_Toc391629717)

[4.2 Restart of system based on hardware watchdog 15](#_Toc391629718)

[4.2.1 Hardware watchdog management 15](#_Toc391629719)

[4.2.1.1 InfoVision TFT platform 15](#_Toc391629720)

[4.2.1.2 Inform TFT platform 15](#_Toc391629721)

[4.3 Restart of system based on ignition 15](#_Toc391629722)

[4.4 Software watchdog 16](#_Toc391629723)

[4.5 Application start and exit control 16](#_Toc391629724)

[5 Communication 17](#_Toc391629725)

[5.1 Local communication on a system 17](#_Toc391629726)

[5.2 Communication between units 17](#_Toc391629727)

[6 Update on a TFT 2.x system 18](#_Toc391629728)

[6.1 Update of system via USB 18](#_Toc391629729)

[6.2 Update of system via FTP 18](#_Toc391629730)

[7 Update of a distributed system 19](#_Toc391629731)

[7.1 Configuration of master unit for update 19](#_Toc391629732)

[7.2 Configuration of slave unit for update 19](#_Toc391629733)

[8 Data Flow between software components 20](#_Toc391629734)

[8.1 Protran and Composer data flow 20](#_Toc391629735)

[8.2 Composer and Renderers data flow 20](#_Toc391629736)

[8.2.1 Multiple Renderers and Composer data flow 20](#_Toc391629737)

[8.3 Data flow in Master-Slave configuration 21](#_Toc391629738)

[9 NLog configuration 22](#_Toc391629739)

[9.1 Log levels 22](#_Toc391629740)

[9.2 Log targets 22](#_Toc391629741)

[10 Time change behavior 22](#_Toc391629742)

[11 Alarms Log 23](#_Toc391629743)

[11.1 alarms.log Configuration 23](#_Toc391629744)

[11.2 Content of alarms.log 23](#_Toc391629745)

[11.2.1 System Reboot 23](#_Toc391629746)

[11.2.2 Application Relaunch 23](#_Toc391629747)

[11.3 Downloading alarms.log 23](#_Toc391629748)

[12 Glossary 24](#_Toc391629749)

**Modification management**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Dept.** | **Modifications** | **State** |
| 0.1 | 2013-10-09 | RAN | SW | Initial version | draft |
| 0.2 | 2013-10-31 | RAN | SW | Updated all chapters | draft |
| 0.3 | 2013-11-11 | RAN | SW | Updated all chapters | draft |
| 1.1 | 2013-11-29 | RAN | SW | Added chapter 10 about behaviour due to time change | draft |
| 2.1 | 2014-02-03 | RAN | SW | Added chapter 11 about alarms.log | draft |
| 2.2 | 2014-06-25 | RAN | SW | Added information about the different Renderers in chapters 2.5, 2.6, 3.2, 8  Modified chapter 11 about alarms.log  Changed all references to TFT 2.0 to TFT 2.x | draft |

**Review**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Dept.** | **Remarks** |
| 0.3 | 2013-11-21 | WES | SW | Minor changes required |
|  |  |  |  |  |
| 1.1 | 2014-01-08 | WES | SW | Reviewed |
|  |  |  |  |  |
| 2.2 | 2014-06-27 | WES | SW | Reviewed, clarified naming (Inform/InfoVision), fixed release server paths |
|  |  |  |  |  |

**Release**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Dept.** | **Remarks** |
| 1.0 | 2013-11-21 | WES | SW | Updated and released |
|  |  |  |  |  |
| 2.0 | 2014-01-08 | WES | SW | Released for TFT System 2.0 |
|  |  |  |  |  |
| 3.0 | 2014-06-27 | WES | SW | Released for TFT System 2.2 |
|  |  |  |  |  |

# Introduction

## Scope

This document is a technical description of the TFT 2.x system. This document is not intended to be a user manual.

The document gives a description of the TFT 2.x system with all the components. It also provides information regarding interaction between the components and how to configure the components so that Gorba internal people can work with TFT 2.x system.

## Intended Audience

This document is written to be understood by Gorba staff familiar with Gorba products. Minimal technical skills are required.

# Product Overview

## System Overview

The TFT 2.x system consists of a combination of different hardware and software components. TFT 2.x system can be configured based on customer requirements. The sections below provide the different hardware configurations possible along with minimum software requirements.

**Unit**(Slave)

**Unit**(Slave)

**Unit**(Slave)

## Different Hardware modes

The TFT 2.x system has two types of units.

### Master-Only mode

This is the most commonly used mode of hardware. In this mode, there is only one Topbox or Compact with its TFT screen. In the master-only mode, the Topbox or Compact will have the following software components.

* System Manager
* Update
* Hardware Manager
* Composer
* DirectX Renderer
* Audio Renderer (if required)
* AHDLC Renderer (if required)
* Protran

### Master-Slave mode

In this mode, there is one master unit connected to several slave units. The master and each of the slave units have their own TFT screen. In the Master-Slave mode, the master and slave units will have the following software components.

On the Master unit:

* System Manager
* Update
* Hardware Manager
* Composer
* DirectX Renderer
* Audio Renderer (if required)
* AHDLC Renderer (if required)
* Protran

On the Slave units:

* System Manager
* Update
* Hardware Manager
* DirectX Renderer

**Unit**(Master)



**LAN**

OBU

**Unit**(Slave)

**Unit**(Slave)

**Unit**(Slave)

### Headless Master-Slave mode

In this mode, there is one master unit connected to several slave units. The master does not have a TFT screen and each of the slave units have their own TFT screen. In the Headless Master-Slave mode, the master and slave units will have the following software components.

On the Master unit:

* System Manager
* Update
* Hardware Manager
* Composer
* Audio Renderer (if required)
* AHDLC Renderer (if required)
* Protran

On the Slave units:

* System Manager
* Update
* Hardware Manager
* DirectX Renderer

### Master unit

The master unit is the main unit in the chain. It is connected to several slave units via LAN. The master unit is the only unit connected to the On-Board computer (OBU). The master unit has the Protran and Composer applications which process the presentation and manage the information from the OBU to transfer the XIMPLE and the information to be rendered to each of the slave units. When the TFT 2.x system is to be updated, the updates for all the slave units are given to the Master unit via USB or FTP. The master unit then transfers the update relevant to the slave to the correct slave unit. The slave unit transfers the feedback and log files from it to the master unit.

### Slave unit

The slave unit contains only the basic software required and the DirectX Renderer application. The slave units interact with the master unit via LAN. Slave units are completely controlled by the master unit. A slave unit updates itself upon receiving an update from the master and informs the master with the feedback of the update and the archived logs from it.

## Hardware of TFT 2.x system

There are different hardware platforms available for use in a TFT 2.x system. The master and slave units can be based on any one of the hardware platforms.

### InfoVision Compact, Quadro, Topbox PC-2 and Topbox Mini

This hardware runs on Windows Embedded 8 Standard. All hardware platforms have several I/O’s, ignition control, display control and a JSON interface for IBIS. The Topbox PC-2 in addition has the Update LED and Button. All other hardware platforms are based on the same motherboard and therefore have the same characteristics.

### Inform Compact and Topbox

The older Gorba Inform TFTs run on Windows XP Embedded. The Inform TFTs have 2 COM ports which also provide I/O’s using the CTS/RTS pins on the COM ports.

## Folder structure on the Topboxes

The folder structure on the Topbox of master or slave unit is as shown below. The folders are placed at the root of the D: drive.

* Config
* Data
* Log
* Progs
* Presentation

### Config folder

The “Config” folder contains the configuration of all the software components used on the system. Each software component has a sub-directory of its name within the Config directory. The sub-directory contains the configuration files for that specific software component. The Config folder is created by the user and can be updated using the Update application. An example of the Config folder with sub-directories for various software components is as seen below.

* AhdlcRenderer
* AudioRenderer
* Composer
* DirectXRenderer
* HardwareManager
* Protran
* SystemManager
* Update

### Data folder

The “Data” folder contains all the data from/of the software components. Each software component has a sub-directory of its name within the Data directory. The Data folder is created and handled only by the software components. This folder is not managed by Update. Some possible data files are persistence files, temporary downloads etc.

### Log folder

The “Log” folder contains the log files from all the software components. It only has one sub-directory, called “Archives”. All the software components are configured to place the log files they generate directly in the Log folder. Based on the configuration, the archived log files for the software components are placed directly within the folder “Archives” in the Log folder. The Log folder is created and handled only by the software components. The “Archives” folder within Log folder is managed by Update.

### Presentation folder

The “Presentation” folder contains the Presentation File (for example, main.im2), with the extension “.im2”. It also contains the folders required by the Presentation File. There will be a tool to create the presentation data and file. The Presentation folder is created by the user and can be updated using the Update application. An example of the folders required by the Presentation File is as seen below.

* Images
* Symbols
* Videos
* Fonts

### Progs folder

The “Progs” folder contains the executables and the DLL’s of all the software components used on the system. Each software component has a sub-directory of its name within the Progs directory. The sub-directory contains the EXE and DLL files for that specific software component. The Progs folder is created by the user and can be updated using the Update application. An example of the Progs folder with sub-directories for various software components is as seen below.

* AhdlcRenderer
* AudioRenderer
* Composer
* DirectXRenderer
* HardwareManager
* Protran
* SystemManager
* Update

## Components

The different software components present in a TFT 2.x system are described in the chapter 2.2.

The basic components, common to both Master and slave units are System Manager, Update and Hardware Manager. These are the essential components for the units to function. Once these essential components are present on a unit, any other required software components can be added to the unit by performing an update of the unit using the Update application.

### System Manager

The System Manager is the main application that controls all the other application on the system. The System Manager can be configured to start, monitor and close applications on the system. This is a software component that must be part of every system. System Manager is usually started automatically by the OS.

The release version of System Manager is available at the location:

[\\gorba.com\daten\Softwareserver\_Release\SW\02\_imotion\02\_TFT\12\_SystemManager](file:///\\\\gorba.com\\daten\\Softwareserver_Release\\SW\\02_imotion\\02_TFT\\12_SystemManager)

The configuration files required for correct operation of System Manager are as follows:

* SystemManager.xml – contains the System configuration, Splash screen configuration and Applications configuration
* Medi.config – contains the configuration of Medi in server mode and configuration required for resource management
* NLog.config – contains the configuration for logging. Refer to chapter 9 for some details about the configuration possibilities.

### Update

The update application handles the update of the applications, presentation and any other part of the system as defined by the update. This is a software component that must be part of every system.

The release version of Update is available at the location:

[\\gorba.com\daten\Softwareserver\_Release\SW\02\_imotion\02\_TFT\14 Update](file:///\\\\gorba.com\\daten\\Softwareserver_Release\\SW\\02_imotion\\02_TFT\\14%20Update)

The configuration files required for correct operation of Update are as follows:

* Update.xml – contains the configuration of the different Clients and Providers that can be used by the Update application, the configuration of applications to be re-launched based on Update and the configuration of the various Visualizations available based on the hardware.
* Medi.config – contains the configuration of Medi in client mode and configuration required for resource management
* NLog.config – contains the configuration for logging. Refer to chapter 9 for some details about the configuration possibilities.

A more comprehensive document about Update application including the configuration is available at

[\\gorba.com\daten\Softwareserver\_Release\Documents\02\_imotion\02\_TFT\14 Update](file:///\\gorba.com\daten\Softwareserver_Release\Documents\02_imotion\02_TFT\14%20Update)

### Hardware Manager

The Hardware Manager is used to control or interact with the hardware of the system. The hardware consists of the IO’s, ignition, LED’s and control of internal features like serial number of the hardware, backlight control etc. All other applications interact with the Hardware Manager when any hardware is required to be accessed by them. This is a software component that must be part of every system.

The release version of Hardware Manager is available at the location:

[\\gorba.com\daten\Softwareserver\_Release\SW\02\_imotion\02\_TFT\13 Hardware Manager](file:///\\\\gorba.com\\daten\\Softwareserver_Release\\SW\\02_imotion\\02_TFT\\13%20Hardware%20Manager%20)

The configuration files required for correct operation of Hardware Manager are as follows:

* HardwareManager.xml – contains the configuration of the hardware being used in the system including I/O’s, accessing buttons and LED’s etc.
* Medi.config – contains the configuration of Medi in client mode.
* NLog.config – contains the configuration for logging. Refer to chapter 9 for some details about the configuration possibilities.

### Protran (Protocol Translator)

The Protran application manages the translation of information from an external source like On-Board Unit (OBU) or CUx into XIMPLE which is transferred to the Composer application via Medi.

The release version of Protran is available at the location:

[\\gorba.com\daten\Softwareserver\_Release\SW\02\_imotion\02\_TFT\09\_ProtocolTranslator](file:///\\\\gorba.com\\daten\\Softwareserver_Release\\SW\\02_imotion\\02_TFT\\09_ProtocolTranslator)

The configuration files required for correct operation of Protran are as follows:

* Protran.xml – contains the configuration about the protocol used, persistence etc.
* Protocol specific configuration file(s) – contain the configuration about the source of information, telegrams, behavior etc.
* Dictionary.xml – contains the basis Protran uses to translate information from the remote PC (On-Board Computer etc.) to XIMPLE.
* Medi.config – contains the configuration of Medi in client mode.
* NLog.config – contains the configuration for logging. Refer to chapter 9 for some details about the configuration possibilities.

A more comprehensive document about Update application including the configuration is available at

[\\gorba.com\daten\Softwareserver\_Release\Documents\02\_imotion\02\_TFT\09 ProtocolTranslator](file:///\\\\gorba.com\\daten\\Softwareserver_Release\\Documents\\02_imotion\\02_TFT\\09%20ProtocolTranslator)

### Composer

The Composer application is one part of the Infomedia which handles the management of the complete presentation. In a master-slave setup, one Composer on the master manages the presentation and provides information to the slaves in order to show the information. One Composer is used to convey information to several Renderers on slaves.

The release version of Composer is available at the location:

[\\gorba.com\daten\Softwareserver\_Release\SW\02\_imotion\02\_TFT\06\_imotion Media Player](file:///\\\\gorba.com\\daten\\Softwareserver_Release\\SW\\02_imotion\\02_TFT\\06_imotion%20Media%20Player)

The configuration files required for correct operation of Composer are as follows:

* Composer.xml – contains the configuration required by the Composer.
* Medi.config – contains the configuration of Medi in client mode.
* NLog.config – contains the configuration for logging. Refer to chapter 9 for some details about the configuration possibilities.

### DirectX Renderer

The DirectX Renderer application is a part of Infomedia which renders the information received from the Composer on the TFT screen.

The release version of DirectX Renderer is available at the location:

[\\gorba.com\daten\Softwareserver\_Release\SW\02\_imotion\02\_TFT\06\_imotion Media Player](file:///\\\\gorba.com\\daten\\Softwareserver_Release\\SW\\02_imotion\\02_TFT\\06_imotion%20Media%20Player)

The configuration files required for correct operation of DirectX Renderer are as follows:

* DirectXRenderer.xml – Contains the configuration of the different screens, the memory cache for images, the mode in which the video shall be rendered etc.
* Medi.config – contains the configuration of Medi in client mode.
* NLog.config – contains the configuration for logging. Refer to chapter 9 for some details about the configuration possibilities.

### Audio Renderer

The Audio Renderer application is a part of Infomedia which renders the information received from the Composer on to an audio device.

The release version of Audio Renderer is available at the location:

[\\gorba.com\daten\Softwareserver\_Release\SW\02\_imotion\02\_TFT\06\_imotion Media Player](file:///\\\\gorba.com\\daten\\Softwareserver_Release\\SW\\02_imotion\\02_TFT\\06_imotion%20Media%20Player)

The configuration files required for correct operation of Audio Renderer are as follows:

* AudioRenderer.xml – Contains the configuration of the different IO’s used for the volume and speaker control, the API to be used for text-to-speech etc.
* Medi.config – contains the configuration of Medi in client mode.
* NLog.config – contains the configuration for logging. Refer to chapter 9 for some details about the configuration possibilities

### AHDLC Renderer

The AHDLC Renderer application is a part of Infomedia which renders the information received from the Composer on to LED sign.

The release version of AHDLC Renderer is available at the location:

[\\gorba.com\daten\Softwareserver\_Release\SW\02\_imotion\02\_TFT\06\_imotion Media Player](file:///\\\\gorba.com\\daten\\Softwareserver_Release\\SW\\02_imotion\\02_TFT\\06_imotion%20Media%20Player)

The configuration files required for correct operation of AHDLC Renderer are as follows:

* AhdlcRenderer.xml – Contains the configuration of the different channels for the RS485 communication, the different LED signs with their address, mode etc.
* Medi.config – contains the configuration of Medi in client mode.
* NLog.config – contains the configuration for logging. Refer to chapter 9 for some details about the configuration possibilities

## Connection between software components

The different software components interact with each other either directly or using some other software components. All the communication between the software components is done using Medi. Below is a diagram explaining the connections between the different software components.

Update

System Manager

Composer

Hardware Manager

Protran

DirectX Renderer(s)

AHDLC Renderer

Audio Renderer

### System Manager ⬄ Update

During an update of an application (including System Manager) or its config, the Update application instructs the System Manager to exit that application in order to perform the update. In case of a System Manager update, it instructs all other applications (except Update) to exit.

### System Manager ⬄ Protran, Composer, DirectX Renderer, Audio Renderer, AHDLC Renderer, Hardware Manager and Update

System Manager and applications Protran, Composer, DirectX Renderer, Audio Renderer, AHDLC Renderer, Hardware Manager and Update are connected to each other. System Manager is responsible for controlling these applications. The details of the control can be referred to in chapter 4.5.

### Protran ⬄ Hardware Manager

The control of I/O’s in Protran is done through the Hardware Manager via Medi. When Inform Topboxes are used, the I/O’s are provided by Protran (CTS/RTS of COM ports) and Protran handles the I/O’s independently. If the system is an InfoVision TFT hardware, the I/O’s are provided by the Hardware Manager to Protran.

When Protran need to set the I/O pin, it sets the pin using the Hardware Manager via GIOoM and Medi.

### Protran ⬄ Composer

The Protran sends the translated information (XIMPLE) from the source to the Composer via Medi. When the Composer is launched or re-launched, it requests Protran for the information via Medi too.

### Composer ⬄ Renderers

The information to be rendered is transferred from the Composer to the different Renderers via Medi. DirectX Renderer informs the Composer in some specific cases, for example when a video has reached the end, via Medi too.

# Hardware Information

## Inputs and Outputs

The TFT 2.x system hardware has several I/O’s and other hardware peripherals depending on the hardware platform.

Some of the hardware peripherals are:

* LED
* Button
* Ignition
* Backlight control

### InfoVision Topbox PC-2 platform

There are 4 digital inputs and 4 digital outputs available on the InfoVision Topbox PC-2 hardware. All the pins can be configured for use in Hardware Manager and may be used by Protran, Renderer, Update or System Manager. The hardware peripherals are:

* Update LED
* Button
* Ignition
* Backlight control

### InfoVision Compact, Quadro and Topbox Mini platforms

There are 6 digital inputs and 2 digital outputs available on the InfoVision Compact, Quadro and Topbox Mini hardware. All the pins can be configured for use in Hardware Manager and may be used by Renderer or System Manager. The hardware peripherals are:

* Ignition
* Backlight control

### Atmel controller on InfoVision TFT hardware

The “Atmel Control” module on the InfoVision TFT hardware handles all the interaction between the main PC and the InfoVision TFT internal hardware. The software components interact directly with the Atmel Control module in order to access the hardware and data like IBIS, system and display information.

### Gorba Inform Topboxes and Compacts

The old Gorba Inform Topboxes and Compacts have 2 COM ports which also provide I/O’s using the CTS/RTS pins on the COM ports. The I/O’s are configured and provided by Protran.

## Software components interaction with Hardware

Different software components interact with the hardware using the Hardware Manager. Some components interact directly with the hardware too. Below are the different interactions between software components and the hardware.

### System Manager interaction with hardware

System Manager Application interacts only with ignition and watchdog on the hardware. System Manager checks the ignition status directly without interacting with the Hardware Manager. System Manager also manages the triggering of the Watchdog directly.

### Update interaction with hardware

Update application interacts with the Update LED on the hardware. This Update LED is used to show the status of an update. Update controls the Update LED directly without interacting with the Hardware Manager.

### Protran interaction with hardware

Protran application interacts with several I/O’s on the hardware but only through the Hardware Manager. If the hardware is a Gorba Inform Topbox, then Protran directly interacts with the hardware and provides those I/O’s through GIOoM.

### DirectX Renderer interaction with hardware

The DirectX Renderer application interacts with the backlight control on the hardware through the Hardware Manager.

### Audio Renderer interaction with hardware

The Audio Renderer application interacts with the speaker enabling output and volume control on the hardware through the Hardware Manager.

# System Management

The System Manager application manages the applications on the TFT 2.x system. All the applications to be managed are configured in the SystemManager.xml.

## System Manager configuration

Each software component to be controlled by the System Manager is configured to:

* enable the application or not
* use the software watchdog of the System Manager for the application. The software watchdog checks for some specific operation of the application are performed regularly. If this operation is not performed regularly, then the application is restarted by System Manager.
* set the path of the application to be managed by the System Manager.
* set the window mode of the application to be minimized, normal, maximized or hidden.
* set the amount of delay System Manager waits before re-launching an application if it was exited.
* System Manager also checks the RAM and CPU for limits (for each application as well as the entire system) and performs an action as determined by the configuration. The action of the System Manager can be re-launching of the application or reboot of the system when the fixed limit is exceeded

## Restart of system based on hardware watchdog

The System Manager application manages the restart of the system along with the application being managed by it depending on the following condition.

### Hardware watchdog management

#### InfoVision TFT platform

The hardware watchdog enabling and setting its reset time are done in the BIOS. Once the hardware watchdog is enabled, the watchdog timer needs to be triggered at regular intervals to avoid a reset of the system due to watchdog timeout. The System Manager is responsible for resetting the watchdog timer at regular intervals. In case there is a problem in the System Manager application and it fails to trigger the watchdog timer, the system is reset by the underlying hardware when the watchdog timer expires.

#### Inform TFT platform

The hardware watchdog on the Gorba platform is always enabled. The System Manager manages the triggering of the watchdog by handling the Port 80 watchdog controller. The System Manager stops the Port 80 service upon startup and then writes to a file at regular intervals. Another service then writes to Port 80 to trigger the watchdog. In case there is a problem in the System Manager application and it fails to write to the file, the system is reset by the underlying hardware.

## Restart of system based on ignition

The ignition is monitored by the System Manager application all the time. As along as the ignition is on, System Manager keeps the applications managed by it, running. When the System Manager detects that the ignition has been turned off, then it waits a configured “Hold time” and then starts the shutdown process. The shutdown process is exiting all the applications it is managing and then shutting the system shown.

When the system is started, System Manager first waits for the ignition status. If ignition is turned off, then System Manager immediately starts the shutdown process else the normal process of starting the configured applications is performed.

## Software watchdog

The System Manager application has a software watchdog available to be used for each application managed by it. The software watchdog can be enabled or disabled for each application. Each application can be monitored for some basic functionality continuously, for example, the rendering loop in Renderer application. As long as the basic functionality is working, the software watchdog is retriggered. When the System Manager detects that the application is not performing the basic functionality, it allows the watchdog to expire which results in a re-launch of the application by the System Manager.

## Application start and exit control

System Manager controls the launch and exit of all the applications configured to be controlled by it. System Manager upon startup then launches each of the configured applications. When an application needs to exit, it informs the System Manager that it requires to exit, System Manager then allows the application to exit. System Manager is informed that the application exited, it then re-launches the said application if it crashed or the application requested for re-launch. When System Manager is exited by the user or another application (except Update), System Manager informs each application to exit first. Once all the applications exit, System Manager also exits. If a reboot of the system is requested, System Manager informs each application to exit and then reboots the system.

# Communication

The communication occurs on two levels. One is locally between the applications on a system. The second is communication between units.

## Local communication on a system

All the applications which communicate with each other use Medi for communication. The System Manager application’s Medi is configured as the “Server”. All the other applications’ Medi are configured as “Client”. Upon startup, the Medi Client on each application connects to the System Manager Medi Server. Henceforth, when an application requires communicating with another application, it transfers information via System Manager. In case an application loses connection to System Manager, the application reconnects to the Medi Server again automatically.

## Communication between units

In a master-slave configuration, all the slave units communicate with the master using Medi. The System Manager application’s Medi on the master unit is configured as the “Server” (see above). The System Manager application’s Medi on the slave unit has an additional configuration as the “Client” with the IP address of master unit specified.

When the master and slave units start, all the slave units’ Medi Clients connect to the master unit’s Medi Server. If the connection is lost, the disconnected slave unit connects automatically to the master unit.

If an application on the master unit wants to communicate with an application on a slave unit, then the application on the master unit sends the information meant for the slave unit to the System Manager. System Manager then sends this information to the particular slave unit’s System Manager. The System Manager on the slave unit then transfers the information to the intended application on the slave unit.

# Update on a TFT 2.x system

Update application allows the user to update a TFT 2.x system independently without any user interaction. A system is updated while running and does not require a reboot. The user can update the entire system, a single application, configuration or data through an update. The update.xml is used to configure the Update application. The update of the system can be performed via USB or FTP. The unit for update is identified by its Hostname. For details about the Update application and its process, refer to the document “TD\_UpdateDocumentation.pdf” available at

[\\gorba.com\daten\Softwareserver\_Release\Documents\02\_imotion\02\_TFT\14 Update](file:///\\\\gorba.com\\daten\\Softwareserver_Release\\Documents\\02_imotion\\02_TFT\\14%20Update).

Currently, a tool “USBUpdateManager” is used to create the updates required. Please refer to the manual “TD\_USBUpdateManager\_UserManual.pdf” to use the update tool which is available at

[\\gorba.com\daten\Softwareserver\_Release\Documents\02\_imotion\02\_TFT\14 Update](file:///\\gorba.com\daten\Softwareserver_Release\Documents\02_imotion\02_TFT\14%20Update).

## Update of system via USB

The entire TFT 2.x system can be updated via USB either unit by unit or as a distributed system. Update of a distributed system is described in chapter 7. Details of update via USB can be referred to in the document “TD\_UpdateDocumentation.pdf”.

## Update of system via FTP

The TFT 2.x system can be updated as a distributed system or in a single unit configuration by receiving the update via FTP file transfer. Details of update via FTP can be referred to in the document “TD\_UpdateDocumentation.pdf”.

# Update of a distributed system

The update of software and data on a distributed system is performed with the master handling the update of all the slave units connected to it. Upon startup, the slave units configured to be updated over a distributed system register themselves to the master unit. Now, the master unit is aware of all the slave units present on the distributed system. When the master unit finds updates for the registered slave units, it sends the correct update to the correct slave unit. The slave units perform the update and send the feedback about the update and the log files (if configured to do so) back to the master unit. The master unit collects the feedback from all the registered slave units and sends it to the source (USB, FTP or backend system).

## Configuration of master unit for update

The update.xml on the master unit is configured to check for updates and subsequently send the updates available for the registered slave units.

The master unit is configured to check for updates available from USB by configuring the “USBUpdateClient”. When a distributed system must be updated, prepare the updates for all the slaves and the master, if required. When the USB stick is connected to the master unit, update application detects the USB stick, waits for the USB detection timeout time to ensure the USB stick is still connected.

The master unit is configured to check for updates available from FTP by configuring the “FTPUpdateClient”. When a distributed system must be updated, prepare the updates for all the slaves and the master, if required. On the master unit, the Update application polls the FTP server for any updates available.

Next, the master unit verifies if there are updates available for itself and any of the registered slave units and downloads the updates.

The master unit is configured to transfer updates to the registered slave units by configuring the “MediUpdateProvider”. The Medi update provider on the master unit ensures that the slave units registered to the master unit receive the update intended for them. It also collects all the feedback from the slave units.

## Configuration of slave unit for update

The update.xml on the slave is configured to register to the master unit to receive updates and send feedback to it.

The slave unit is configured to register to the master unit to receive updates from it by configuring the “MediUpdateClient”. The Medi update client on the slave unit ensures that the slave unit registers to the master unit and receives the updates sent by the master to it. It also sends the feedback from the slave to the master unit.

# Data Flow between software components

The software components between which the main data flow occurs are:

* Protran
* Composer
* DirectX Renderer
* Audio Renderer
* AHDLC Renderer

Protran

Composer

DirectX Renderer

Ximple

Request for data

Items to render

End of video (pool) notification

Audio Renderer

AHDLC Renderer

Items to play

Items to render

## Protran and Composer data flow

Protran provides information to the Composer using the XIMPLE. Protran also receives request to resend the XIMPLE from the Composer when the Composer starts.

## Composer and Renderers data flow

The Composer creates the set of items to render based on the presentation file (\*.im2) and the XIMPLE received from Protran or other source. The Composer sends the items in broadcast mode to all the Renderers available. The Composer sends updates to the screen too to the Renderers.

All Renderers upon start-up send a request for the right screen resolution to the Composer. The Composer then sends the items for the Renderer based on the correct screen resolution.

The DirectX Renderer notifies the Composer when a video playing from a pool ends.

### Multiple Renderers and Composer data flow

When there are multiple Renderers with different resolutions for example, the Composer sends the items with different resolutions required to all the Renderers in broadcast. The Renderers render the items based on the correct resolution for it.

## Data flow in Master-Slave configuration

Master

Slave

Register to Master

Update of application and data

Feedback about update

Notifications

Data for rendering

The data flow in the Master-Slave configuration has been explained in chapter 5.2. The figure above explains the various information flows that occur in the Master-Slave configuration. First the registration of the Slave to the Master occurs. Then the Master sends the information required to the slave. The Slave sends notifications to the Master when required. The Master handles the transfer of all updates required to the Slave. The Slave sends the acknowledgement and the feedback to the Mater about the updates and other information like log files etc.

# NLog configuration

The logging using NLog can be configured to suit the different requirement of the user.

## Log levels

The different log levels possible with Nlog are:

* Off – this means that logging is disabled.
* Fatal – logs all logs upto the fatal level
* Error – logs all logs upto the error level
* Warn – logs all logs upto the warn level
* Info – logs all logs upto the info level
* Debug – logs all logs upto the debug level
* Trace – logs all logs upto the trace level

## Log targets

There are two possible log targets configured for every application:

* Console – logs everything to the console based on the log level
* File – logs everything to the file system at the location specified in the configuration. The logging to file system is to be used only for maintenance purposes.

# Time change behavior

The change in the system time can occur due to the change of daylight saving time or manual change of system time by the user or by Composer (based on the values in the “System” table. The change in system time does not affect the time management behavior for timers which have only an interval configured. Actions like polling every “x” seconds, starting an application after “x” seconds will function correctly irrespective of the time jump. It means that if an event is supposed to start in x seconds and the system time changes then the event will still start after x seconds exactly. If there is an operation configured to start at the exact time, for example, “23:00:00” (HH:MM:SS), then this is affected by the change is system time. All software applications which have a configuration of exact time handle the change is system time to ensure the operation takes place at the correct configured time.

It is very important to note that due to the limitations of Windows Embedded 8 Standard, systems that are updating their time automatically need to be configured to UTC time zone (see Hardware Manager documentation). Failure to do so will result in the system to change its time by one hour on every reboot as soon as the daylight saving boundary is passed (i.e. DST to standard time or vice versa).

# Alarms Log

One of the log files always enabled for logging on a TFT system is the “alarms.log” file. The alarms.log file is created by the System Manager and is saved in the Log folder of the TFT system.

## alarms.log Configuration

The alarms log is configured in the NLog.config of the System Manager application. The name of the logger in NLog.config for the alarms logging must be “Gorba.Motion.SystemManager.Core.Alarms.AlarmService”.

## Content of alarms.log

The alarms log file contains the different alarms raised by the System Manager when it handles different applications. The log file contains a list of time stamp and the associated alarm that was raised at that time. The two alarms that are available are:

### System Reboot

The System Reboot alarm is logged whenever a system reboot occurs. The information available with this alarm is the name of the system and reason for the reboot separated by semi-colons in the same order.

Below is couple of examples of the System Reboot alarm log entries

* 2014-06-19;13:50:29.377;System;SystemRestart;Unknown;"First boot"
* 2014-06-19;12:00:00.783;System;SystemRestart;User;"Reboot after Requested: Reboot: From TFT-43-AE-D8:HardwareManager: Updated system settings"
* 2014-06-19;13:58:44.804;System;SystemRestart;Unknown;"Reboot after SystemShutdown: Windows shutdown"

### Application Relaunch

The Application relaunch alarm is logged whenever System Manager relaunches an application it handles. The information available with this alarm is the name of the application being relaunched and reason for the relaunch separated by semi-colons in the same order.

Below is couple of examples of the Application relaunch log entries

* 2014-06-19;13:52:31.105;Application;ApplicationRelaunch;SystemBoot;"HardwareManager: "
* 2014-06-02;15:13:12.211;ApplicationRelaunch;"Update";"Requested: From TFT-43-AE-D8:Update: Self-update completed"
* 2014-06-13;17:43:49.307;Application;ApplicationRelaunch;SoftwareUpdate;"HardwareManager: From TFT-43-AE-D8:Update: Update done"

## Downloading alarms.log

The alarms.log files are downloaded to either a USB stick or uploaded to a FTP server by the Update application along with the other log files created the TFT applications.

# Glossary

|  |  |
| --- | --- |
| CTS | Cleat To Send. This is a signal line of a serial port |
| FTP | File Transfer Protocol over TCP/IP |
| GIOoM | Gorba I/O over Medi protocol (pronounced “Guillaume”); protocol for reading digital inputs and writing to digital outputs between different applications or different units |
| I/O | Inputs/Outputs |
| LED | Light Emitting Diode |
| Medi | Message Dispatcher; communication protocol used by all TFT 2.x applications |
| NLog | The logging platform used to perform all the software logging on a Gorba system |
| OBU | On-Board Computer is the external source of the data/information for the TFT 2.x system |
| OS | Operating System |
| Protran | PROtocol TRANsalator is the software component which handles the conversion of the information received from an external source like the On-Board Computer into XIMPLE. |
| RTS | Request To Send. This is a signal line of a serial port |
| Unit | Unit in the TFT 2.x system is the InfoVision or Inform TFT |
| XIMPLE | Extended Information Management Protocol |