

|  |  |
| --- | --- |
|  | **imotion** |
|  |  |
|  | TFT2 |
|  |  |
|  | Functionality Tests and Results |
|  |  |
|  | Technical Description |
|  |  |
|  | Version: 0.8 |
|  | State: Draft |
|  | Classification: Internal use only |
|  | Author: RAN |
|  | Creation date: 2013-01-18 |
|  | Repository: $/Gorba/Main/MotionInfomedia/Documents/Functionality Tests and Results |
|  | Gorba AG  Sandackerstrasse  9245 Oberbüren  Switzerland |

**Table of contents**

[1 Introduction 4](#_Toc349899054)

[1.1 Scope 4](#_Toc349899055)

[1.2 Intended Audience 4](#_Toc349899056)

[2 Product Overview 4](#_Toc349899057)

[2.1 System Overview 4](#_Toc349899058)

[3 Functionality Test cases with results 4](#_Toc349899059)

[3.1 Text Rendering Options 4](#_Toc349899060)

[3.2 Smooth news Ticker 5](#_Toc349899061)

[3.2.1 Smooth ticker concept 5](#_Toc349899062)

[3.2.2 Smooth ticker test cases with results 5](#_Toc349899063)

[3.3 Smooth video 6](#_Toc349899064)

[3.3.1 Source Video 6](#_Toc349899065)

[3.3.2 Conversions 7](#_Toc349899066)

[3.3.3 Results 8](#_Toc349899067)

[3.3.4 Technologies Used 8](#_Toc349899068)

[3.3.5 Useful Screenshots 9](#_Toc349899069)

[3.3.6 Conclusions 10](#_Toc349899070)

[3.3.7 Further Ideas 10](#_Toc349899071)

[3.4 Integrated Logging 10](#_Toc349899072)

[3.5 (Almost) VDV300 compliant IBIS interface 11](#_Toc349899073)

[3.6 Ethernet timing 11](#_Toc349899074)

[3.6.1 UDP fast transmission and reception 11](#_Toc349899075)

[3.6.2 Fast response behaviour 11](#_Toc349899076)

[3.7 General CPU/memory utilisation 12](#_Toc349899077)

[3.8 “Streaming” to Ethernet slaves 13](#_Toc349899078)

[4 XNA Framework Analysis 13](#_Toc349899079)

[4.1 Introduction 13](#_Toc349899080)

[4.2 Analysis of Compatible Xna framework versions 13](#_Toc349899081)

[4.2.1 List of requirements on Topbox with PM800 13](#_Toc349899082)

[4.3 Analysis of Xna Framework rendering capabilities 13](#_Toc349899083)

[4.3.1 Xna Framework 4.0 13](#_Toc349899084)

[4.3.2 Xna Framework 3.0 13](#_Toc349899085)

[4.4 Conclusion 14](#_Toc349899086)

**Modification management**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Dept.** | **Modifications** | **State** |
| 0.1 | 2013-01-20 | RAN | SW | First version with section 3.1 | draft |
| 0.2 | 2013-02-01 | WES | SW | Added test results of text testing | draft |
| 0.3 | 2013-02-05 | WES | SW | Added test summary for video testing | draft |
| 0.4 | 2013-02-12 | RAN | SW | Added documentation about concept for smooth ticker. Added repository place to cover page. Modified System overview with the information about screen cloning feature. | draft |
| 0.5 | 2013-02-15 | COS | SW | Extended the chapter 3.2 “Smooth video” | draft |
| 0.6 | 2013-02-18 | RAN | SW | Reorganized chapter 3.1 and 3.2. Added details to chapter 3.1 Text rendering options. Modified a point in chapter 3.3 | draft |
| 0.7 | 2013-02-18 | RAN | SW | Added chapter on Ethernet timing | draft |
| 0.8 | 2013-02-28 | RAN | SW | Added chapter on XNA Framework analysis | draft |

**Review**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Dept.** | **Remarks** |
| 0.5 | 2013-02-18 | RAN | SW | Editing of chapter 3.2 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Release**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Dept.** | **Remarks** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Introduction

## Scope

This document is a compilation of different functionality tests for TFT2 with the obtained results.

## Intended Audience

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

# Product Overview

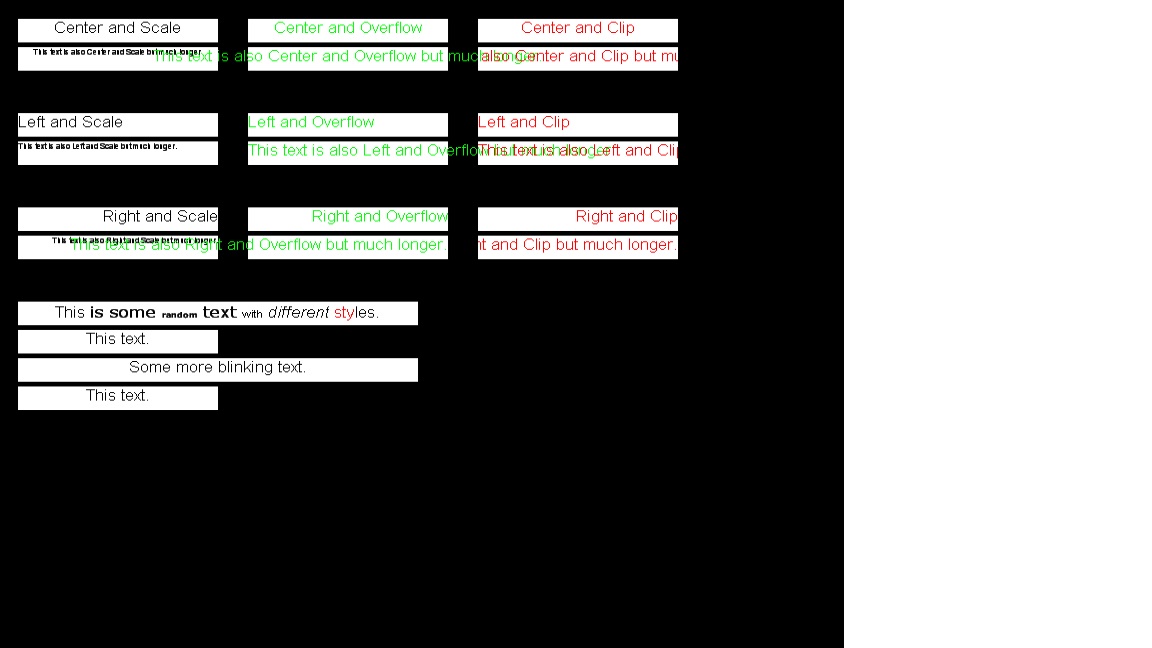
## System Overview

All tests are performed on Topboxes with either PM800 or Atom. The hardware used for the tests are specified in the test case. All the functionality tests have been performed with the screen “**cloning**” feature disabled using the IEGD tool.

# Functionality Test cases with results

## Text Rendering Options

All text rendering is done with the screen “700” of the Infomedia 2 test project using Arial size 20 with 22 texts, some with formatting and blinking / alternating. Below is the screen shot of the screen “700” of Infomedia 2 test project.



The 2 modes of rendering are Fullscreen Window Mode and Fullscreen Exclusive Mode. A presentation is the complete set of all items to be rendered on the screen.

Fullscreen Window Mode – The presentation is shown in a window extending to the complete screen.

Fullscreen Exclusive Mode – The presentation is shown in full screen (taking ownership of the screen).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Frames per Second | PM800 Fullscreen Window Mode | PM800 Fullscreen Exclusive Mode | ATOM Fullscreen Window Mode | ATOM Fullscreen Exclusive Mode |
| Font AntiAliased\* | 20 / 29 | 34 / 37 | 60 / 60 | 60 / 60 |
| Font Default\* | 20 / 29 | 34 / 37 | 60 / 60 | 60 / 60 |
| Font NonAntiAliased\* | 23 / 30 | 35 / 39 | 60 / 60 | 60 / 60 |
| Font ClearType\* | 20 / 20 | 30 / 31 | 31 / 31 | 48 / 49 |
| Font ClearTypeNat.\* | 20 / 20 | 30 / 31 | 31 / 31 | 48 / 49 |
| Font Proof (=Draft)\* | 27 / 30 | 36 / 39 | 60 / 60 | 60 / 60 |
| GDI | 30 | 42 | 60 | 60 |
| Mesh | 8 | 10 | 15 | 16 |

\* = value on the left without Sprite, value on the right with Sprite.

As seen above, the frame rate improves in the Fullscreen Exclusive Mode. When the end-user wants to play a video with other text or pictures in the presentation, it is better to use Fullscreen Window Mode.

A summary of the results from the table above is that, for PM800, Font NonAntiAliased and Font Proof (=Draft) produces the best results. The text rendered using these 2 methods will look different from the standard text rendering methods. Text rendering using GDI shall not be used.

With Atom, any of the methods of text rendering mentioned in the table above is feasible.

## Smooth news Ticker

### Smooth ticker concept

Smooth ticker is an extension of text rendering with the option to scroll the text. Two parameters, Overflow and ScrollSpeed can be set with options/values to create a ticker text. Set the parameter “Overflow” to “Scroll” and set a ScrollSpeed between 0 and +/-Value (pixels/second) for a ticker text. The sign in the scrollspeed value indicates the direction of the ticker. ScrollSpeed with “-“sign indicates that the text will scroll from right to left (RTL) and “+”sign indicates that the text will scroll from left to right (LTR).

When the overflow property of a text is set to scroll, the scroll offset is calculated using the scrollcounter (incremented every millisecond, width of the text and width of the rectangle within which the text is rendered.

### Smooth ticker test cases with results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Parameters | HW | FullScreenWindowed mode | FullScreenExclusive mode |
| 2 tickers running parallel (LTR + RTL) | Scroll speed = 60 | PM800 | 50fps  Smooth tickers | 65 fps  Smooth tickers |
| 10 tickers with different scroll speeds |  | PM800 | 35 fps  Smooth tickers | 43 fps  Smooth tickers |
| Ticker with image | Scroll speed = 60 | PM800 | 48 fps  Smooth tickers | 55 fps  Smooth tickers |
| Ticker with video | Scroll speed = 60  DirectShow rendered | PM800 | 25 fps  Smooth tickers and video | 30 fps  Smooth tickers and video |
| Ticker with text, image and video | Scroll speed = 60  DirectShow rendered | PM800 | 25 fps  Smooth tickers with image and video | 30 fps  Smooth tickers with image and video |
| Ticker over fullscreen video | Scroll speed = 60  DirectShow rendered | PM800 | 4 fps  Smooth video and ticker | 7 fps  Smooth video and slow ticker |
| Ticker with different BBCode | Scroll speed = 60 | PM800 | 48 fps  Smooth tickers | 60 fps  Smooth tickers |

## Smooth video

In order to understand the limits about the TFT System 2.0 about rendering video, we did extensive tests using several codecs and containers with videos having different resolution and playing them in different hardware. Using several combinations of containers and codecs for video with different resolutions has revealed the capabilities of the current hardware.

### Source Video

First of all, let’s start to explain something about the videos used.  
We needed to have an “official video” in high resolution as the base for each manipulation/conversion. The official video can be downloaded from this path:

<http://mirrorblender.top-ix.org/peach/bigbuckbunny_movies/big_buck_bunny_1080p_h264.mov>

Its dimension is around 700 MB, and therefore it’s not stored in any Gorba’s server. Its duration is more or less 10 minutes, bit rate 9283 Kbps, 24 FPS, width 1920 pixels, height 1080 pixels, MPEG-4 as container and AVC1 as codec.

Starting from it, we selected some portions in order to have a smaller video but always in high definition. The selected portions are:

* from the 00:11 to 00:15
* from the 00:23 to 00:34
* from the 03:13 to 03:18
* from the 06:59 to 07:04
* from the 08:35 to 08:40

We took those portions in order to select the “zones” with fading effects, a lot of colors, scrolling texts, rapid changes in the scenes, etc…

The result of the video editing is:

N:\User\COS\TFT2.0\_Videos\VIDEO\_IM2.avi  
  
The video above is the **source** of all the converted video. All its properties can be viewed in the following image:

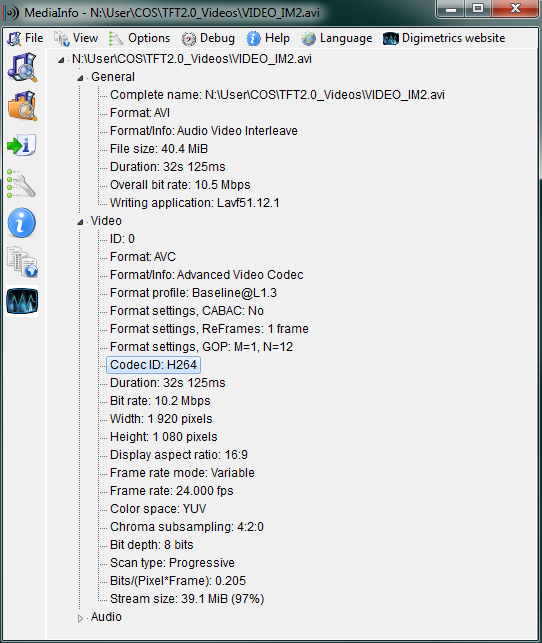


Figure 3‑1 Information about the source video.

### Conversions

The goal of the test was to know what can be played well on the devices (PM800 and Atom) and therefore we needed more videos. After some research we defined the following combinations of container and codec:

|  |  |  |
| --- | --- | --- |
| **Videos (container/codec/resolution)** | | |
| AVI | WMV2 | 1920\*1080 |
| 1440\*900 |
| 640\*360 |
| MP42 | 1920\*1080 |
| 1440\*900 |
| 640\*360 |
| MPEG-4 | AVC1 | 1920\*1080 |
| 1440\*900 |
| 640\*360 |
| MPEG2 | MPG2 | 1920\*1080 |
| 1440\*900 |
| 640\*360 |

Table 1 Combinations of codec/containers

All the videos above (organized by container) can be found here:

N:\User\COS\TFT2.0\_Videos\VIDEO\_TESTS\_IM2

To obtain those videos, we used the software “Any Video Converter[[1]](#footnote-1)”. Each converted video has a high bitrate: the minimum bit rate is 4222 kbps for the AVI/MP42 (640x360), the maximum is around 10.4 Mbps. Using that conversion tool, was not possible to obtain something better.

It has also to be noted that the conversion doesn’t affect the size of the result. The source video has a size of circa 40 MB and this value is also mostly maintained in the output (for the video with less resolution, the output size is obviously smaller).

### Results

The extensive test results can be found in the document “Video Codecs.xlsx”. Below we give a short summary of the findings:

1. It is very important to disable the screen “**cloning**” feature using the IEGD tool. The frame rates in some of our tests improved by up to 40%.
2. Playing **videos below other items** (using “DirectShow” instead of “DirectXWindow”) leads in most cases to worse video quality, only a MPG2 640x360 video played reasonably well on the PM800 in this mode
3. None of the videos really played well on the **PM800**, the best quality was achieved using MPEG 4.2 at 640x360
4. Rendering videos on the **1920x540 widescreen** using the Atom resulted in bad quality
5. The **Atom** rendered most videos very well (using “DirectXWindow” mode) with almost no loss of quality
6. Playing **two videos in parallel** is basically impossible
7. In general the **best results** were achieved using the MPEG 4.2 codec
8. On PM800 the MPG2 codec is rendered **partially transparent** with “DirectXWindow” mode, therefore this codec is not usable.

The file “Video Codecs.xlsx” can be found here:

$/Gorba/Main/Motion/Infomedia/Documents/Video Codecs.xlsx

### Technologies Used

To render video have been used two options:

* DirectXWindow
* DirectShow

In order to play a video below items, the end user must use DirectShow to render the video. DirectShow renders a video onto a texture. The quality of the video rendered using DirectShow is worse compared to DirectXWindow. To play a Video Top Most over Picture, DirectXWindow is a better choice.

### Useful Screenshots

In the file

$/Gorba/Main/Motion/Infomedia/Documents/Video Codecs.xlsx

are mentioned the hardware setups and configurations used during the steps.  
Just to help the reader, here below will be shown some screen shots in order to explain graphically what it means “Video Below Items” and “Video Top Most over Picture”.

A video below some items (pictures and/or texts) is well represented in the following image:

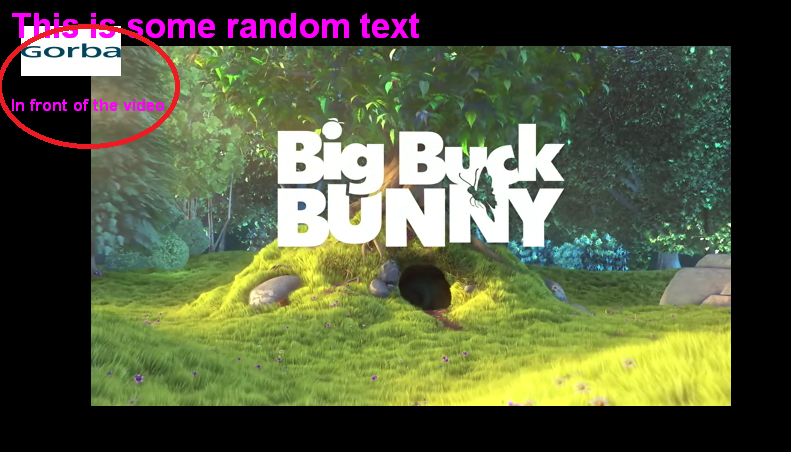


Figure 3‑2 Example of video below items

In the red circle on the upper left corner is selected a portion of the video with a picture and a text rendered above the video itself. This test was required to understand if InfoMedia 2.0 can render items on top of a video, as could reasonably happen in case of a video and a ticker.

A video above some items (pictures and/or texts) is well represented in the following image:



Figure 3‑3 Example of video above items

In the red circle on the upper left corner is selected a portion of the video with a picture and a text rendered below the video itself. This test was required to understand if InfoMedia 2.0 can render video on top of a texts or pictures, as could reasonably happen in case of a video and a perlschnur.

### Conclusions

On a PM800, only a video with MPG2 640x360 can be used for “Video below items”. On an Atom, only videos with a resolution of 640x360 with different containers and codecs are usable for “Video below items”.

During our tests we always had FRAPS running. A detailed article about the usage of FRAPS to calculate the FPS performances is available at this URL:  
  
<http://ixbtlabs.com/articles2/video/fraps.html>  
  
The testers of that article discourage the usage of FRAPS only for analysis involving games. In our cases, just playing video, FRAPS can be used without having bad impacts on the result.

Finally, compared to Windows Media Player, the performances of InfoMedia 2.0 are somehow lower: Windows Media Player plays also the video slowly, with small jumps and not with a great quality, but overall the result is surely better.

### Further Ideas

In order to reach some more improvements, can be interesting to evaluate if it’s possible to stop render all the stuffs in background when is playing a video in full screen. This could save CPU consumption and lead therefore to a better quality of the video itself.

## Integrated Logging

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test number | Test Case | Parameters | HW | Result | Comments |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## (Almost) VDV300 compliant IBIS interface

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test number | Test Case | Parameters | HW | Result | Comments |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Ethernet timing

### UDP fast transmission and reception

The concept is to test the timing with UDP transmission and reception. A file of 500 bytes with a 2 byte counter and timestamp is transmitted over a UDP. The test setup consists of 2 PM800’s, one for sender and one for receiver.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | HW | Result (Average Rx Jitter) | Result (Maximum Rx Jitter) | Comments |
| Without Infomedia 2 running | PM800 | 1ms | 23ms | Within specified limits |
| With Infomedia 2 running | PM800 | 2ms | 19ms | Within specified limits |

### Fast response behaviour

The concept is to test the fast response behaviour of the UDP on PM800. The UDP sender on one PM800 sends a special status request telegram (ABCDEFGH) every second. The second PM800 upon receiving this telegram sends an answer (12345678) back to the UDP sender. The total time between transmission and reception on the sender PM800 is calculated on the sender.

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | HW | Total time (Average) | Comments |
| Without Infomedia 2 running | PM800 | 2ms | Within specified limits |
| With Infomedia 2 running | PM800 | 100ms | Within specified limits |

## General CPU/memory utilisation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test number | Test Case | Parameters | HW | Result | Comments |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## “Streaming” to Ethernet slaves

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test number | Test Case | Parameters | HW | Result | Comments |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# XNA Framework Analysis

## Introduction

Xna framework was analyzed as an alternative to Managed DirectX to render Infomedia 2.0. The analysis was focused on rendering text, images and videos. The analysis was performed in several steps.

* Requirements on PM800 to get Xna working
* Requirements with associated software for Xna to render text, images and videos.

## Analysis of Compatible Xna framework versions

The first point of analysis was to check which version of Xna framework is compatible with.NET Framework 2.0. Xna 3.0 is compatible with .NET Framework 2.0 and Windows Xp Sp3. Xna 3.0 also requires Shader 1.1. The graphics chipset on Topbox with PM800 does not support Shader 1.1.

### List of requirements on Topbox with PM800

To use Xna Framework 3.0 on Topbox with PM800, the following support is required.

1. .NET Framework 2.0 and Windows Xp Sp3
2. Graphics chipset with Shader 1.1

## Analysis of Xna Framework rendering capabilities

Different version of the Xna Framework was analyzed for their rendering capabilities of text, images and videos. The sections below provide the results and capabilities of each version of Xna Framework respectively. All version of Xna framework require the fonts and images to be pre-rendered during application development time using the “Content pipeline” concept of Xna. Font and images cannot be loaded easily during runtime. This is a disadvantage and a deterrent to using Xna Framework for rendering Infomedia 2.0.

### Xna Framework 4.0

Xna Framework 4.0 requires .NET Framework 4.0 to run. This framework supports rendering of text, images and videos. The hardware of the Topbox does not support running Xna framework on it.

### Xna Framework 3.0

Xna Framework 3.0 requires a minimum of .NET Framework 2.0 with Windows Xp Sp3 to run. This framework supports rendering of text and images. To render videos, 3rd party API DirectShow is required. The hardware of the Topbox does not support running Xna framework on it.

## Conclusion

The conclusion of the analysis is that Xna Framework 3.0 or higher can be used in the future to render Infomedia 2.0, provided a powerful graphics processor and upgraded software requirements are satisfied on the Topbox.

1. \\Softwareserver\Release\SW\02\_imotion\02\_TFT\00\_Software Tools\09 AVC Free [↑](#footnote-ref-1)