MOBILE3DTV

# Second version of the terminal device

Nikolay Daskalov ■ Ivan Pushkarov ■ Nikolay Vladimirov

MOBILE3DTV

**MOBILE3DTV**

**Project No. 216503**

# Second version of the terminal device

*Nikolay Daskalov, Ivan Pushkarov, Nikolay Vladimirov*

**Abstract:** This report describes the hardware and software design and implementation of the second version of the terminal device. At that stage the goal has been twofold: first, to start developing a novel general framework for integration of all software components needed for the normal functioning of the final device, including the architecture and the resource control, second, to integrate the DVB-H front-end to the development platform. This integration consists of hardware interfacing of the selected DVB-T receiver module and design and implementation of all the required software components. Namely, appropriate OS, Linux SPI driver, driver support for the DVB-T module, DVB-T signal processing software, appropriate player, flexible framework in which all the software components will be implemented. The goal of this device version is to help the team to verify the selected interfaces both SW and HW. The other goal is to evaluate all the SW components, and to do some measurements of the performance for future optimization.

**Keywords:** 3DTV, mobile video, OMAP, DVB-H

# Executive Summary

The hardware implementation of the second version of the terminal device has been accomplished. At that stage the form factor has not been targeted. All the HW components which will be used in the final version of the terminal device, were gathered together. Namely, the processing platform, the auto-stereoscopic LCD and the DVB-T/H receiver module.

Needed software components concerning the DVB-T/H module were requested and delivered from the module vendor. All the additional documentation, as “Porting guide” and “Reference design”, were also delivered.

The module was first tested on Linux PC. This helped the team to validate the non-platform-specific software, and to evaluate the basic performance of the module. This was also important for outlining the plan for the platform porting, driver development and all additional applications.

In order to accomplish the target version, new HW components were developed. PCB board for interfacing the DVB-T/H module and the OMAP3430 EVM, also compatible with the new releases of the OMAP3430 SDP, namely Zoom-1 and Zoom-2 were developed.

From SW perspective, the driver for the DVB-T/H module was developed and debugged; modifications were made on the SDK provided by the vendor, so to suit the development platform. Additional SPI test application, needed for measurements and validation of the interface, and additional flexible framework for lighter implementation of all the SW components on the device were developed.

# Table of Contents

# 1 Framework development

## *1.1 Framework overview*

For the implementation of all the additional software components of the final device MMS has developed a new framework. In Figure 1.1, a diagram explaining the architecture and the resource control is shown. Our Framework is based on an efficient inter-processor communication [1], connecting the system the General Purpose Processor (GPP) and the Digital Signal Processor (DSP).

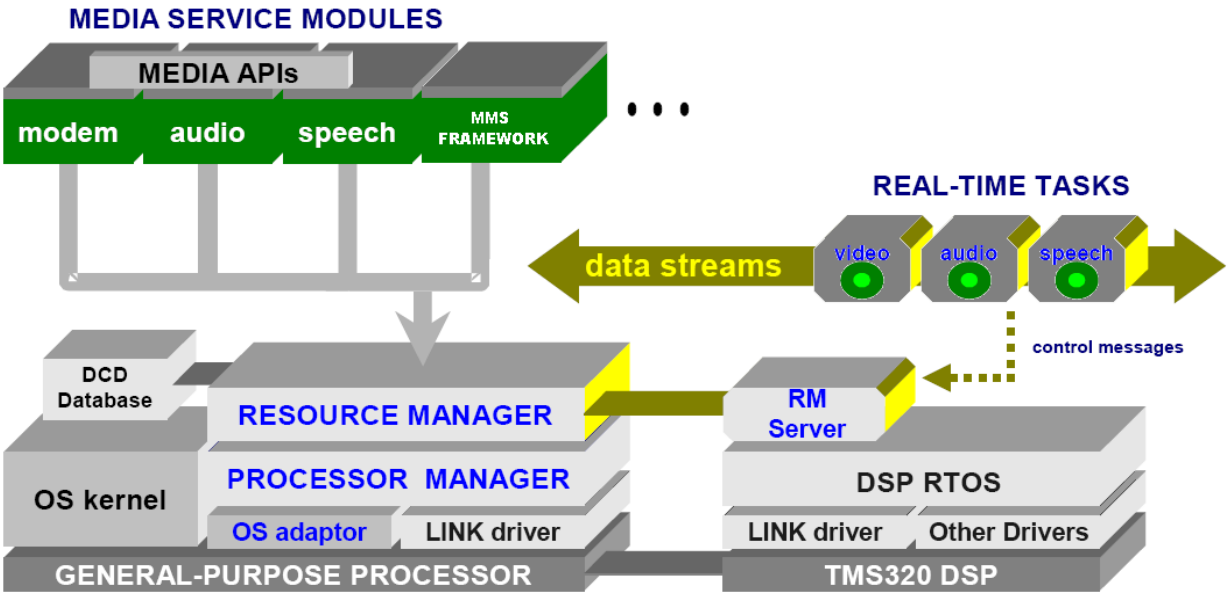



**Figure 1.1**

Applications working on the GPP make calls into media service modules or drivers. These modules and drivers are designed to use the DSO/BIOS bridge API to manage DSP resources.

The Resource Manager has the following responsibilities:

- dynamically instantiating DSP resources, meeting the allocation requests,
- monitoring DSP resources
- dynamically loading DSP codes as needed
- implement policies for managing DSP resources when conflict requests appear

The Resource Manager is on top of the Platform Manager which has the following responsibilities: statically loading a base code image to the DSP, starting and stopping the DSP, and implementing data streaming.

The Platform manager is on top of the GPP OS adaptation layer and the DSP ”link” Driver used for low-level interprocessor communication with the DSP. Configuration information and the packaged DSP content for the platform are stored in the DCD (Dynamic Configuration Database)

On the DSP side, the communications between the BSP/BIOS and the GPP is via the Host “link” Driver. On top is the RM (Resource Manager) Server. Its main duties are: dynamically create, execute and destroy DSP processing nodes. This is done under the RM control. Routing the

messages between the GPP and individual nodes, altering task priorities, and responding to RM configuration commands and status queries, is also done by RMS. A dedicated data stream is used to send commands from the RM to RMS. Another dedicated stream is used to send responses back to the RM.

Each DSP task node is a separate execution thread, which runs on the DSP that implements control or signal processing algorithms. Communication between different task nodes and the GGP is via short fixed-length messages and/or device-independent I/O.

## *1.2 Data transfer types*

There are two types of data transfer:

- Control data transfer
- Processing data transfer

Figure 1.2 illustrates the difference between the processing data and the control data.

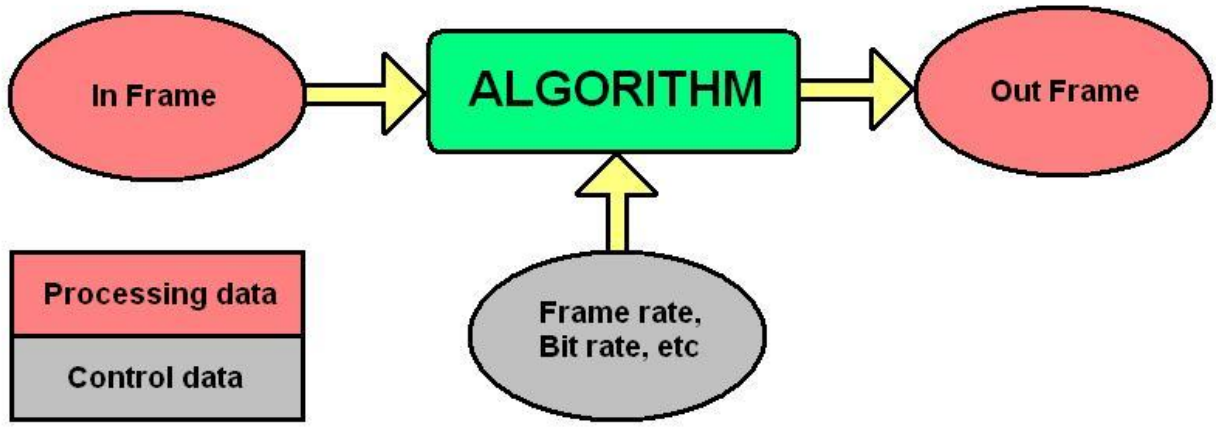



**Figure 1.2 Difference between processing and control data**

The data that goes for processing through an algorithm is the processing data. It can be: image, frame, sound sample, etc or part (portion) of these.

Another type of data is the control data. It is used to control/setup the algorithm with the appropriate values/parameters, like: frame rate, bit rate, contrast, etc.

The two types are transferred between the GPP and the DSP in two different ways.

### 1.2.1 Control data transfer

The transfer of the control data between the GPP and the DSP is done by using shared memory region. It uses small packets up to 10kB.

We have two memory spaces one for the DSP and one for the GPP. They have so called “shared memory”. This memory is placed in different addresses in these two memory blocks. This shared memory is visible from both sides, from the GPP memory and the DSP memory.

The concept of the shared memory region is illustrated in Figure 1.3.

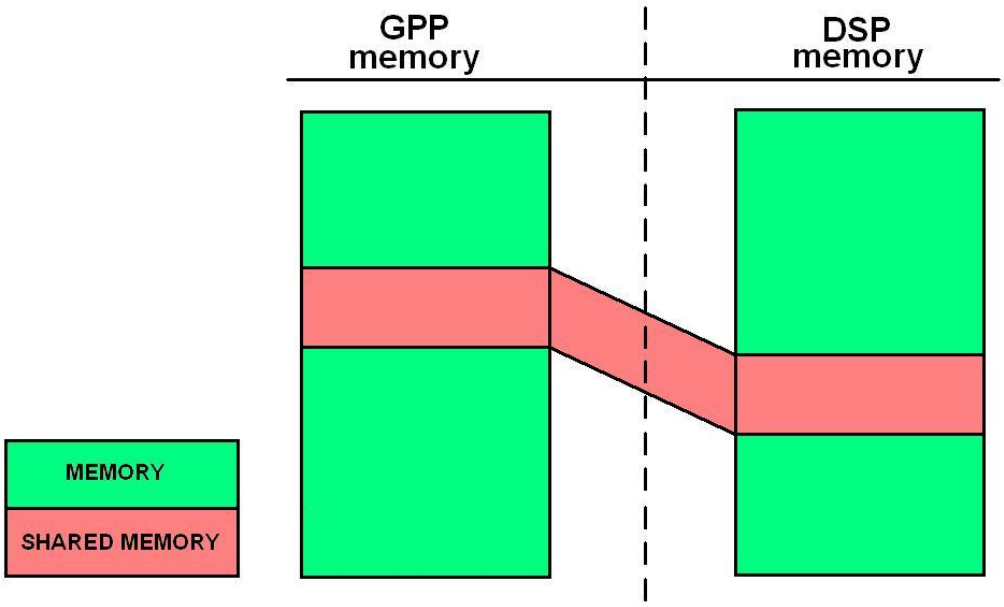



**Figure 1.3 Control data transfer**

The process of sequence of the transfer is explained below:

First a memory is allocated in the GPP shared memory. Then this allocation is filled with data. Next a message containing the address in the GPP memory is sent to the DSP. At the DSP shared memory the address is converted and the data is accessed. The data is passed for processing. Then acknowledge is sent back to the GPP share memory. This way the data is transferred from the GPP, to the DSP.

When the data is transferred from the DSP to the GPP the sequence is different:

Memory is allocated for the data is the GPP memory. Message is sent to the DSP with the address of the allocation, the data is filled in. Then acknowledge is sent back to the GPP, and the data is accessed and passed for further action.

### 1.2.2 Processing data transfer

The transfer of the processing data between the GPP and the DSP is done by using virtual memory regions in the GPP and DSP memory spaces.

Figure 1.4 illustrates the processing data transfer using virtual memory.

First two blocks of memory are allocated in GPP and DSP memories, and the addresses are transcoded. This is done by the DSP bridge. The GPP block is then filled, and a message containing the address in the GPP memory is sent to the DSP. The message is received and the data is taken for processing. Acknowledge is sent back to the GPP, when the data is processed, and the GPP retrieves it.

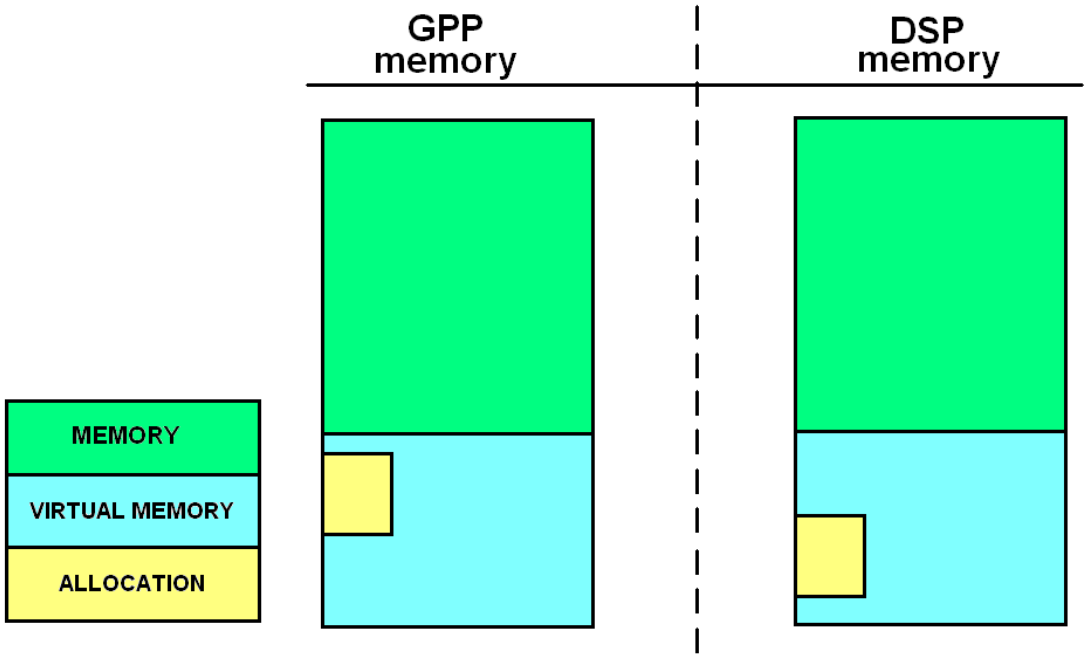



**Figure 1.4 Processing data transfer**

# 2 Overview of the DVB-H stack

DVB-H is based on DVB standards used in Europe for terrestrial and satellite DTV transmission. Low – power mode is available for battery – powered devices. DVB-H is used for handheld terminal devices (lightweight, battery powered)[2] [3] [4]. These devices require special features from the transmission system which serves them:

- Possibility of repeatedly turning the power off to some parts of the receiver. This reduces the power consumption.
- To ensure the receiver location change from one transmission cell to another without losing the DVB-H service.
- Flexibility to be used in various bandwidth and transmission bands etc.

In the DVB-H new technological elements are used, namely time slicing and FEC coding. [9] [10] [11] [12]

## *2.1 Time slicing*

This technology is used in order to reduce the power consumption of the receiving terminal by approximately 90-95%. Time slicing is mandatory for DVB-H. Figure 2.1 illustrates the time slicing.

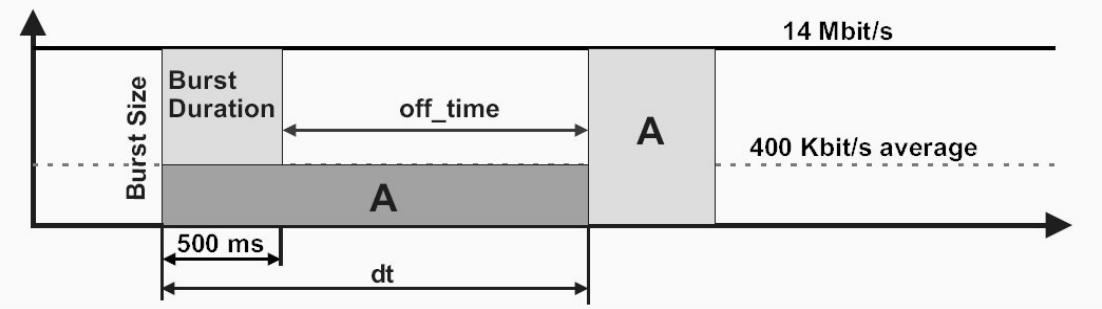



**Figure 2.1 Time slicing illustration**

The signal is sent in bursts at higher transfer speed. During the time between the bursts the receiver is powered off. This reduces the device power consumption. Figure 2.2. shows the difference in the data transmission in DVB-H and DVB-T.

Numerous DVB-H services are sent in packets. This allows sending the info in burst mode. The data is not sent continuously. The DVB-T data is sent continuously. Figure 2.2 shows the difference between DVB-H and DVB-t Transmission [13].

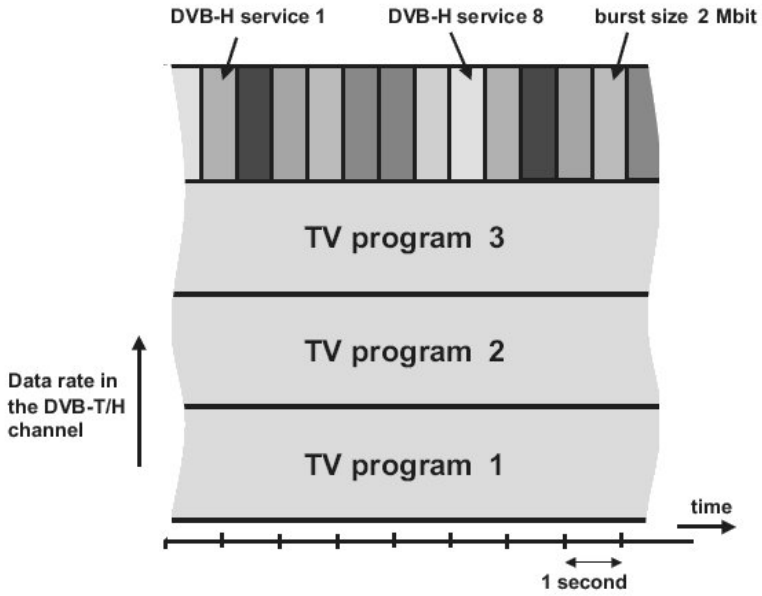



**Figure 2.2. Difference between DVB-T and DVB-H**

## 2.2 FEC coding

FEC stands for Forward Error Correction - this is a system where the sender adds redundant data to the code called error correction code. The code is added on a bit-stream level. The amount of the added redundant code is measured in levels marked by fractions. FEC = 7/8 means that one-eight of the total bits number will be occupied by the extra error-protection code [5].

The flowchart of transmitting and receiving processes is shown in Figure 2.3

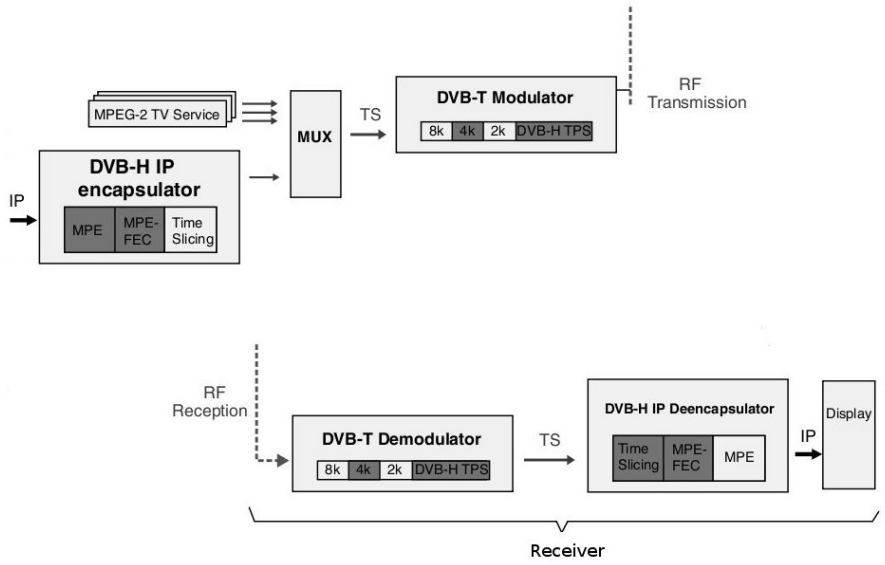



**Figure.2.3. Transmitting and receiving**

At the receiver side there are two main processes. Demodulation and De-encapsulation. These processes and their sub-processes will be described in Section 2.1 for the selected DVB-H receiver module.

A diagram of the DVB-H stack is shown in Figure 2.4

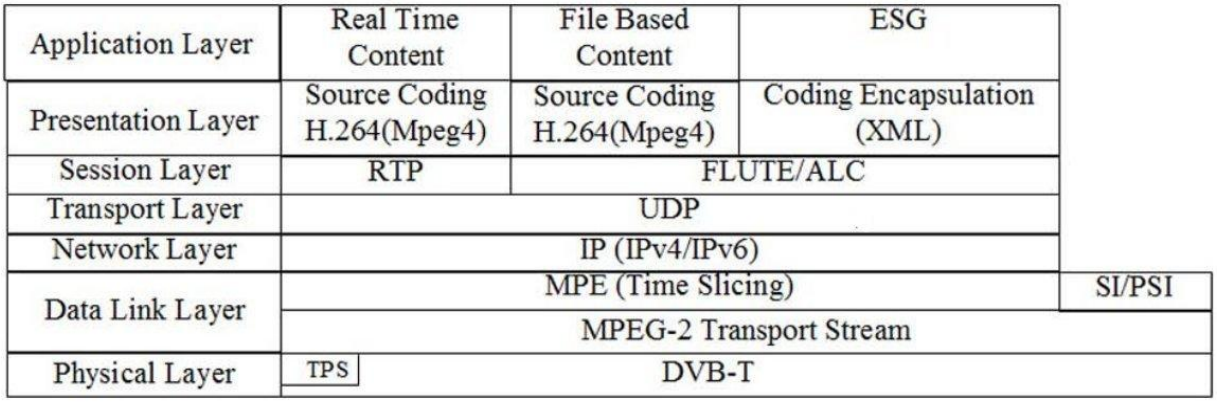



**Figure 2.4 DVB-H stack**

DVB-H is a multimedia transmission system which transports various multiple applications and files with different formats such as video and audio streaming, file transfer, electronics service guide, XML or HTML data [6].

At the application layer there are Real Time Content, File based content, ESG (Electronic Service Guide). The real time content (TV broadcasting) is sent via RTP (Real Time Transfer Protocol). The non-real time content like files and ESG is sent via File Delivery Over Unidirectional Transport / Asynchronous Layer Coding (FLUCE/ALC).

At the Data Layer there are Multi Protocol Encapsulation (MPE) and Service information/ Program Specific Information (SI/PSI) [7] [8].

Figure 2.5. illustrates the MPE-FEC table formation.

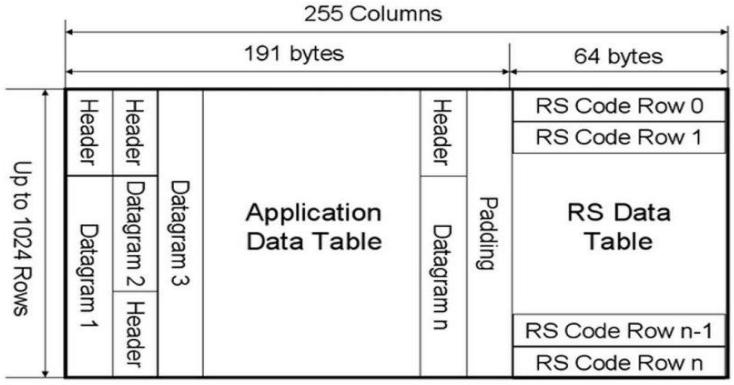



**Figure 2.5 MPE-FEC table formation.**

The IP datagrams are stored in the “Application Data Table”, and the Reed Solomon encoder is applied to each Application Data Table row to produce 64 byte FEC code words. This content is transmitted column by column.

# 3 DVB-H front-end

## *3.1 Basic elements of the DVB-H stack as implemented in the DibCom chip*

The selected chip is DIB9080H delivered by DibCom. Basic block diagram of the selected DVB-T/H receiver module is shown in Figure 3.1.

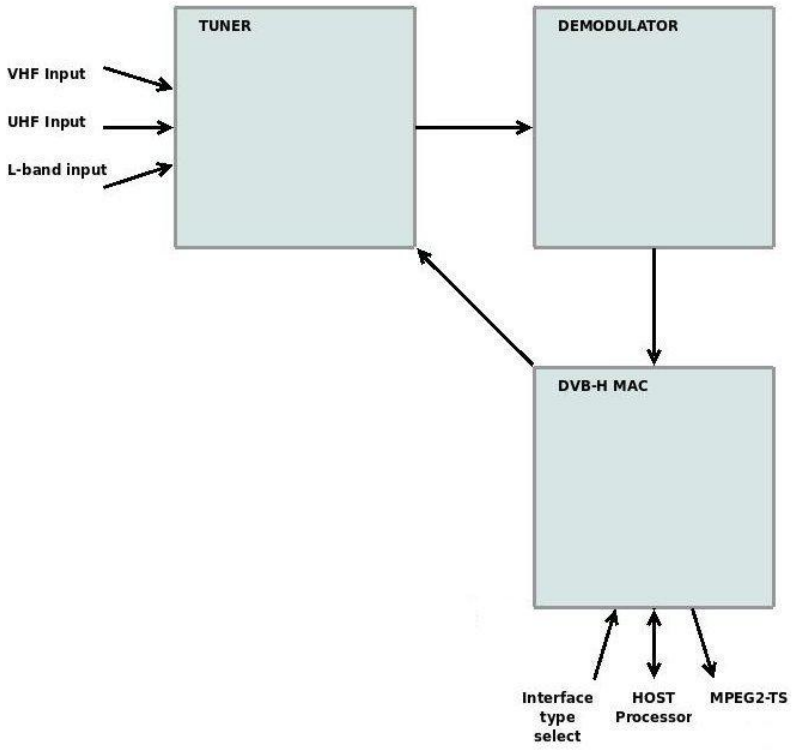
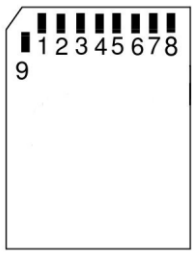



**Figure 3.1**

The module can be logically separated in three parts TUNER, DEMODULATOR, DVB-H MAC. There are three inputs for the module VHF, UHF and L-band, entering the tuner. There is also another input controlling the interface type, entering the DVB-H MAC. There is one bidirectional transfer line between the DVB-H MAC and the Host processor. This is the control communication between the devices. The output is MPEG2-TS coming out the DVB-H MAC.

## *3.2 Interfacing the DibCom DVB receiver to OMAP 3430 platform*

The mode selected for this implementation is SPI. The SDIO connector scheme is shown in Figure 4.2.




For interfacing the board, one SD card slot and two connectors are used. The connectors are SEAM-20-03.0-S-10-2-A-K and MIT-038-02-F-D-K. Mating connectors are available on the OMAP3430SDP board.

The picture of DibCom module interfaced to OMAP3430SDP through the interface board is shown in Figure 4.4.

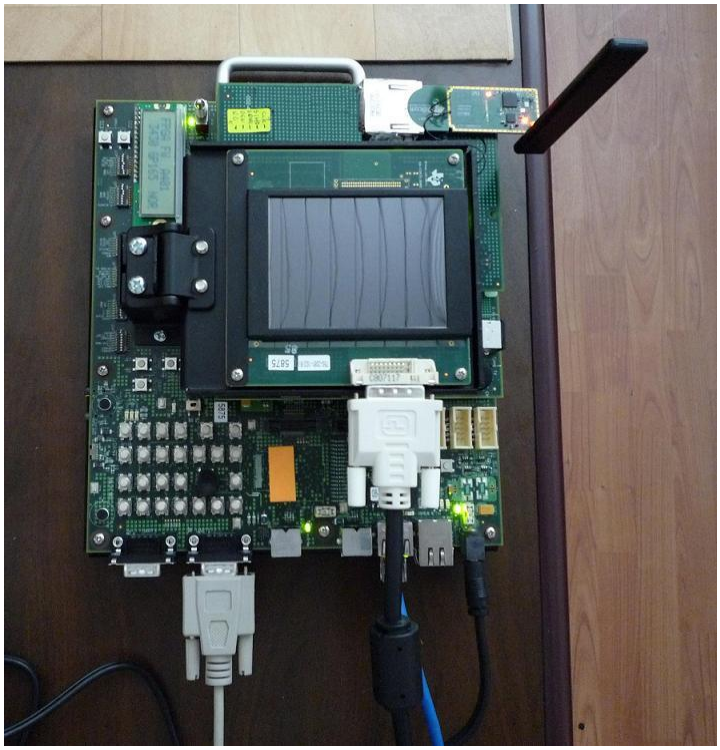


**Figure 4.4.**

The system shown on the picture of Figure 4.4 is AP3430SDP EVM running Linux OS. DibCom module is hardware and software implemented and DVI support is enabled for the NEC display.

From hardware point of view this is the second version of the terminal device with all hardware modules integrated. No form factor has been addressed at that time.

# 4 Conclusions

All hardware modules have been properly interfaced to the development platform. All additional hardware was developed, tested and integrated. The hardware implementation of the SPI interface was verified.

The Dibcom software was successfully ported and compiled for the OMAP3430 platform and was tested with DVB-T transmission. We were able to receive MPEG-2 transport stream in UDP packets. De-capsulation of the TS, and decoding related with error resilience schemes is being developed. Collaboration regarding this development has been established with WP2 and WP3.

# 5 References

[1] Texas Instruments, "DSP/BIOS™ Bridge Programming Guide", 23 January 2008

[2] Lars-Ingermar Lundström, "Understanding Digital Television", Elsevier Inc, 2006

[3] Amitabh Kumar, "Mobile TV: DVB-H, DMB, 3G Systems and Rich Media Applications", Elsevier Inc, 2007

[4] John Arnold, Michael Frater, Mark Pickering, The University of New South Wales, ADFA, Canberra, ACT, Australia, "Digital Television – Technology and Standards", John Wiley & Sons, Inc, 2007

[5] Daniel Minoli, "IP Multicast with Applications to IPTV and Mobile DVB-H", John Wiley & Sons, Inc, 2008

[6] Xiaodong Yang, "Handover in DVB-H – Investigation and Analysis", Springer-Verlag Berlin Heidelberg, 2008

[7] Borko Furht, Syed Ahson, "Handbook of Mobile Broadcasting: DVB-H, DMB, ISDB-T, AND MEDIAFLO", Taylor & Francis Group, LLC, 2008

[8] Hervé Benoit, "Digital Television – Satellite, Cable, Terrestrial, IPTV, Mobile TV in the DVB Framework", Dunod, 4th edition, 2006

[9] ETSI, EN 300 468, "Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems" V1.5.1 (2003-01)

[10] ETSI, TR 102 469, "Digital Video Broadcasting (DVB); IP Datacast over DVB-H: Architecture" V1.1.1 (2006-05)

[11] ETSI, EN 300 744, "Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television", V1.5.1 (2004-06)

[12] ETSI, EN 302 307, "Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications" V1.1.1 (2004-06)

[13] Wikipedia, "DVB-H", http://en.wikipedia.org/wiki/DVB-H

# Mobile 3DTV Content Delivery Optimization over DVB-H System

MOBILE3DTV

MOBILE3DTV - Mobile 3DTV Content Delivery Optimization over DVB-H System - is a three-year project which started in January 2008. The project is partly funded by the European Union 7th RTD Framework Programme in the context of the Information & Communication Technology (ICT) Cooperation Theme.

The main objective of MOBILE3DTV is to demonstrate the viability of the new technology of mobile 3DTV. The project develops a technology demonstration system for the creation and coding of 3D video content, its delivery over DVB-H and display on a mobile device, equipped with an auto-stereoscopic display.

The MOBILE3DTV consortium is formed by three universities, a public research institute and two SMEs from Finland, Germany, Turkey, and Bulgaria. Partners span diverse yet complementary expertise in the areas of 3D content creation and coding, error resilient transmission, user studies, visual quality enhancement and project management.

For further information about the project, please visit **www.mobile3dtv.eu**.

**Tuotekehitys Oy Tamlink**
Project coordinator
**FINLAND**

**Tampereen Teknillinen Yliopisto**
Visual quality enhancement,
Scientific coordinator
**FINLAND**

**Fraunhofer Gesellschaft zur Förderung der angewandten Forschung e.V**
Stereo video content creation and coding
**GERMANY**

**Technische Universität Ilmenau**
Design and execution of subjective tests
**GERMANY**

**Middle East Technical University**
Error resilient transmission
**TURKEY**

**MM Solutions Ltd.**
Design of prototype terminal device
**BULGARIA**

MOBILE3DTV project has received funding from the European Community's ICT programme in the context of the Seventh Framework Programme (FP7/2007-2011) under grant agreement n° 216503. This document reflects only the authors' views and the Community or other project partners are not liable for any use that may be made of the information contained therein.