IMPLEMENTATION OF AN IMAGE RECOGNITION ALGORITHM ON THE DM6446 DAVINCI PROCESSOR

P. Ayyalasomayajula [1], *S. Grassi* [1], *N. Deurin* [2], *P.-A. Farine* [1] *and T. Guéguen* [2]

[1] Ecole Polytechnique Fédérale de Lausanne, Institute of Microengineering, Electronics and Signal Processing Laboratory
EPFL STI IMT ESPLAB, Rue A.-L. Breguet 2, 2000, Neuchâtel, Switzerland
phone: + (41) 32 718 34 25, fax: + (41) 32 718 34 02, email: pradyumna.ayyalasomayajula@epfl.ch
web: http://esplab.epfl.ch

[2] Epicard S.A., Rue Fernand-Dubois 3, 1890 St-Maurice, Switzerland
web: www.epicard.ch

ABSTRACT

We are developing an Alternative and Augmentative Communication (AAC) portable device called PictoBar which is used in speech rehabilitation therapy. PictoBar recognizes barcodes and images, such as pictograms and pictures. Then it plays a sound message associated with the recognized barcode or image. This paper describes the development of the image recognition algorithm and its implementation using Codec Engine framework on a DM6446 DaVinci processor.

1. INTRODUCTION

The field of electronic aid for disabled people is growing constantly and many innovations are added every year. In particular, there is an increasing need for electronic aids in Alternative and Augmentative Communication (AAC). Further, the use of picture symbols (pictograms) has proved successful in clinical studies to improve the communication in people with mild-to-severe speech-impairments. Communication boards, i.e., arrangements on a physical support of pictograms that convey graphical messages, are widely used for this purpose.

The Swiss Foundation for Rehabilitation technology [1] together with speech therapists has identified the need for a portable, rugged and user friendly device, able to recognize a pictogram and play a sound or phrase associated to it. This device will be used to improve speech therapy using AAC.

In this paper we describe the development of such a device, called "PictoBar", using Texas Instruments DaVinci technology. Emphasis is placed in the description of the development of image recognition algorithm and its implementation on a DM6446 DaVinci processor, using Texas Instruments Codec Engine framework. The paper is organized as follows. The PictoBar device is described in Section 2. The development of the image recognition algorithm is explained in Section 3. Section 4 describes the software framework used for the DSP implementation. Section 5 explains the DSP implementation. Current status and Future work are presented in Section 6. Finally conclusions are drawn in Section 7.

2. DEVICE DESCRIPTION

PictoBar is a portable and rugged AAC device with image (pictures and pictograms) and barcode recognition capabilities, used as communication aid and for therapy of speech-impaired people. Once an image or barcode is recognized, PictoBar plays a pre-recorded sound message, associated to

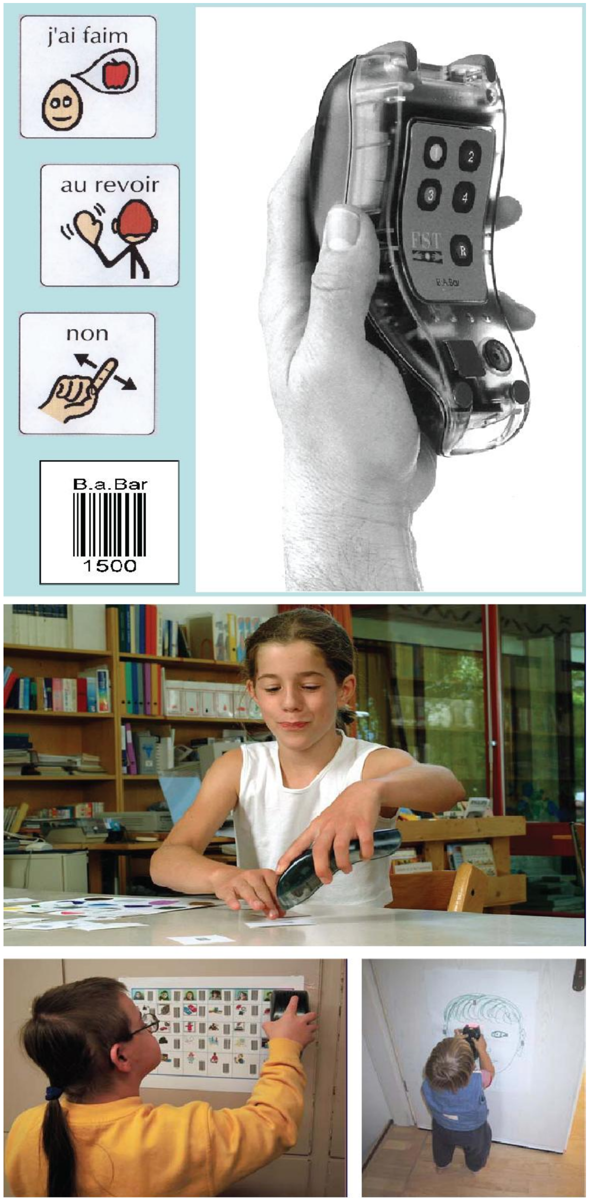

Figure 1: B.A.Bar, a present generation AAC device and examples of its utilization

the recognized image or barcode. In order to be recognized, an image (picture or pictogram) must have been previously stored into the device database. PictoBar is an improved version of B.A.Bar [2], a present generation AAC device which recognizes barcodes and plays sound messages associated to them. Figure 1 shows B.A.Bar and its utilization.

2.1 Device hardware

The hardware block diagram of PictoBar is shown in Figure 2. Main components are:

- *Micro-camera:* used to acquire the images or barcodes printed on communication boards. The camera is a fixed focus OV7675 from Omnivision [3] with VGA resolution (640x480 pixels).
- *Processing unit:* is a DaVinci DM6446 dual core processor from Texas Instruments [4]. This processor has an ARM926EJ-S core and a fixed point C64x+ DSP core. The application runs on the ARM core, featuring user interface, barcode recognition algorithm, peripheral control and overall system management. The image recognition algorithm runs on the DSP core.
- *Audio I/O Interface:* consisting of an Audio Integrated Circuit (IC TLV320AIC26) [5], a microphone for audio acquisition and a speaker for audio playback. The AIC contains AD/DA converters used in 16-bit mode, 16 kHz sampling frequency.
- *USB Interface:* used to connect PictoBar to an external PC, for firmware updates and to train new images into PictoBar database.
- *Memory:* PictoBar contains built-in memory (Flash NAND and DDR2-SDRAM) as well as detachable memory (SD card). The FLASH NAND is used to store the programs. The SD Card is used to store the database containing the JPEG compressed images and their pre-extracted features, together with the voice messages associated to each image.

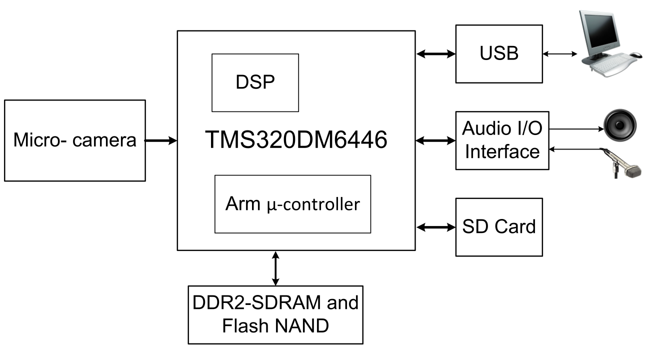

Figure 2: Hardware block diagram of PictoBar

3. IMAGE RECOGNITION ALGORITHM

Figure 3 shows the block diagram of the image recognition algorithm, developed at ESPLAB, that is used in PictoBar. The constitutive blocks of the recognition algorithm are explained as follows:

- *Border Recognition, Reconstruction and image Preprocessing (BRRP):*
  This block is used to align and crop the input "query image" i.e., image to be recognized which is acquired with the camera. The images in the communication board are, by convention, printed with a rectangular border enclosing its content. The enclosing border is detected using line detection, based on modified Hough transform. The query image is then aligned and cropped. Then, image pre-processing such as histogram stretching is applied.
- *Color Density Circular Crop (CDCC) preselection algorithm:*
  To decrease the complexity while keeping a good performance, image recognition is done in two stages. In the first stage we use a less complex but also less robust algorithm called "Color Density Circular Crop" (CDCC). This algorithm searches the database and preselects the 50 images which are closest to the query image. In the second stage, a more robust but computationally more expensive algorithm called "DCT phase match" (DCTPM) is used. This algorithm searches the closest match to the query within the reduced database of the 50 images preselected in the first stage.
  The CDCC preselection algorithm searches the closest images in the database using as feature the color proportions within concentric circular zones of the images. Feature calculation requires edge detection to determine these zones. As the features are pre-calculated and stored in the image database, the CDCC algorithm is low in complexity. Additionally, this algorithm is robust to rotation, translation and scaling, but is sensitive to variations in lighting.
- *DCT Phase Match (DCTPM):*
  The base recognition algorithm, which is referred to as DCT phase match [6], searches for the best match of the query image into a database using correlation on the DCT phase of the 8x8 blocks of the images. This algorithm is compatible with the JPEG compression standard. The diagram of the DCT phase match algorithm is shown in Figure 4. This algorithm is accurate and robust to lighting variation but computationally expensive. To reduce its complexity, DCT phase match is only performed on the reduced database of images preselected by the CDCC algorithm.

3.1 Camera Settings and image database for testing

The database of pictograms used during development is a subset of 1500 pictograms randomly selected from the Picture Communication Symbols set [7]. We refer to this set of 1500 pictograms as the "clean database" and use it as the reference database in which the query image is searched. Res-

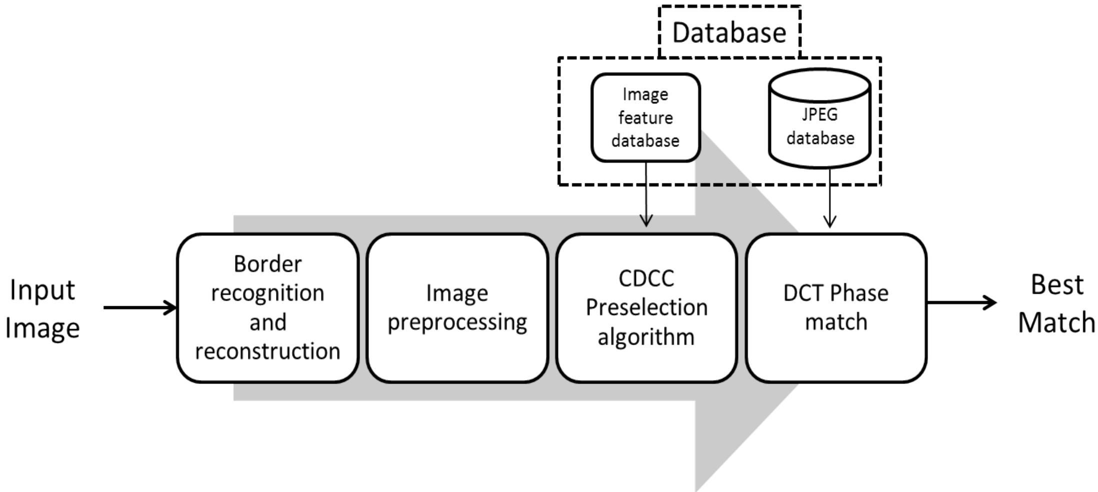

Figure 3: Image recognition algorithm

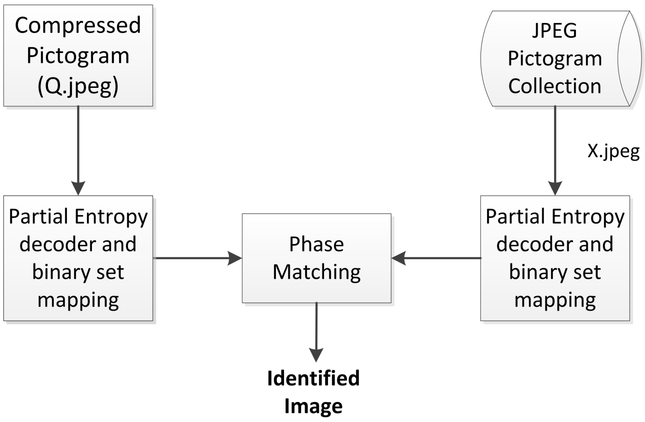

Figure 4: DCT phase match algorithm

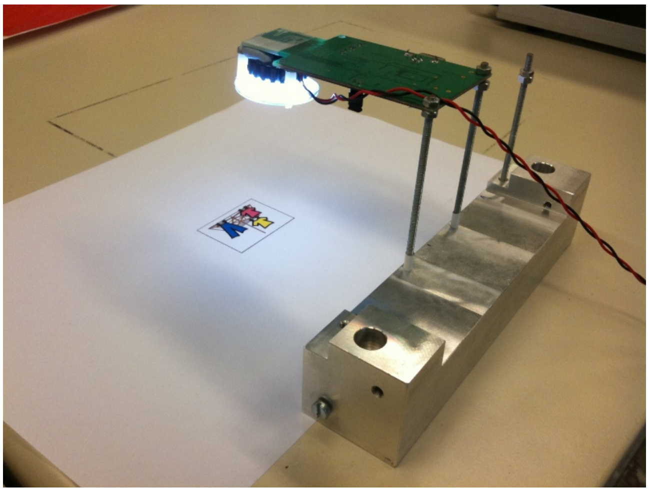

Figure 5: Camera setup with LED crown

olution of this database was adjusted to be 320x240 pixels (QVGA).

To reproduce the conditions in the final device, the fixed focus OV7675 camera was fixed at a distance of 9.2 cm from the image. Under this condition, a VGA resolution acquired image corresponds to a rectangle of 6 x 8 cm. A LED crown was mounted on the camera to provide uniform lighting. This is required as the CDCC preselection algorithm is not robust to lighting variation. A subset of 50 representative pictograms, selected from the 1500 of the "clean database" was printed on paper with size of 3 x 4 cm. To provide visual reference for manual alignment and cropping, two enclosing borders (3 x 4 cm and 6 x 8 cm) were printed as shown in Figure 6. These pictograms were acquired with the camera, from MATLAB, manually aligned using the printed borders as visual help, and cropped to 3 x 4 cm (320x240 pixels), thus removing both borders. These 50 acquired images constitute the "real condition database" and are used as query images for the tests.

The camera setup including the LED crown is shown in Figure 5, and an example of printed pictogram for testing is shown in Figure 6.

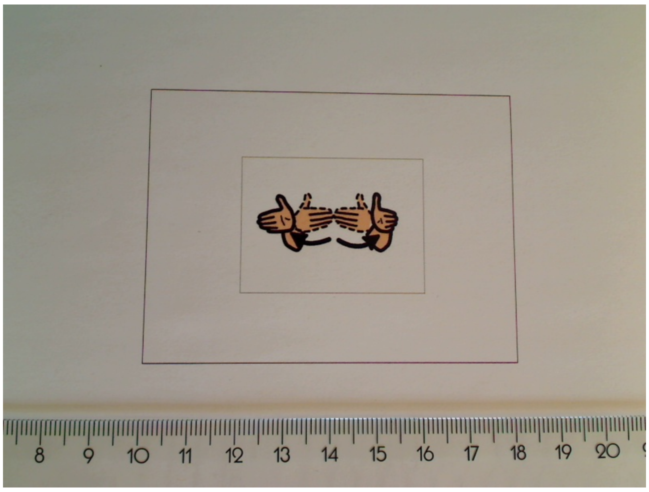

Figure 6: Example of printed pictogram

3.2 Algorithm development and testing

The algorithm was implemented and tested in MATLAB using Image processing, Signal processing and Image acquisi-

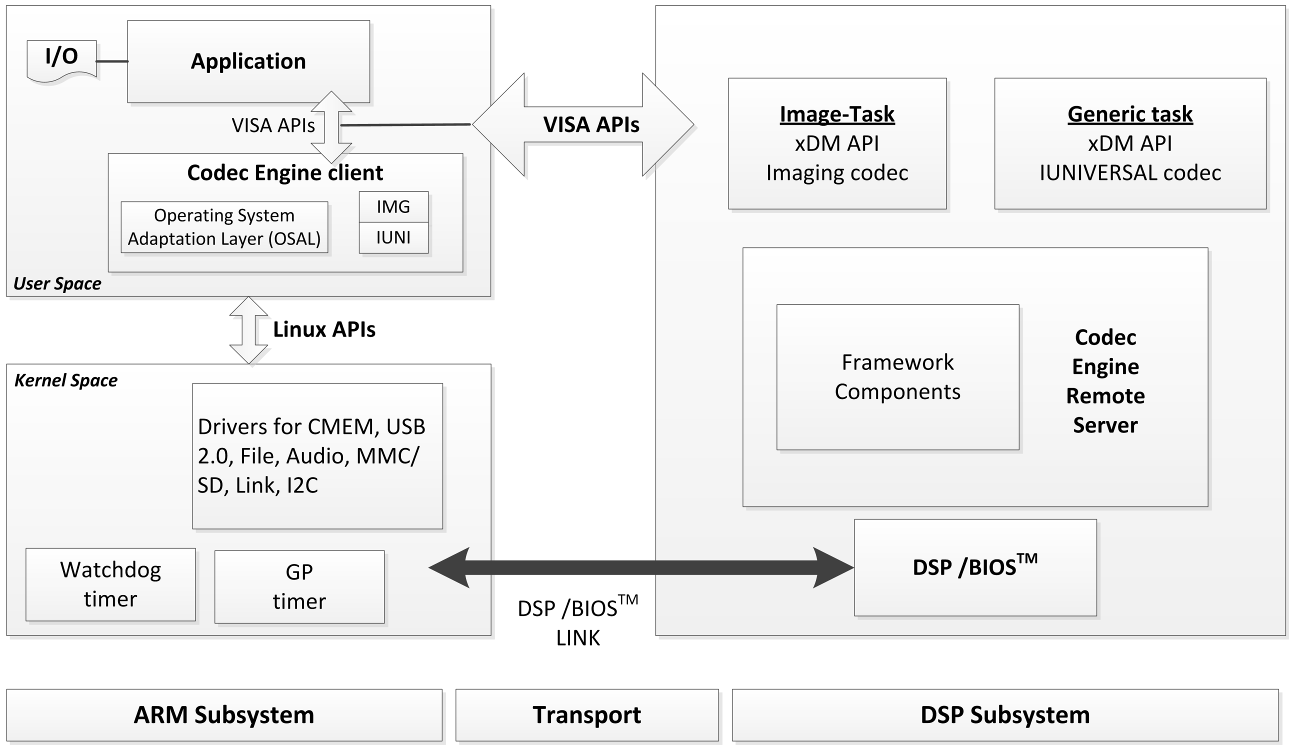

Figure 7: Codec Engine software components

tion toolboxes. The algorithm was tested initially with query images from the "clean database". After this first validation, the algorithm was tested with query images acquired from the camera (from the "real condition database").

4. SOFTWARE FRAMEWORK FOR DSP IMPLEMENTATION

The DM6446 contains an ARM9 core and a fixed point C64x+ DSP core. The ARM9 core uses Embedded Linux as RTOS. The application running on the ARM core features user interface, barcode recognition algorithm, peripheral control and overall system management. Additionally, the ARM application controls the remote execution of algorithms on the DSP core using Codec Engine framework as explained below. DSP core uses DSP / BIOS^{TM} [8] as real-time kernel. The ARM and DSP communicate with each other through DSP / BIOS^{TM} LINK.

To implement the recognition algorithm we use the Codec Engine (CE) [9] framework whose software components are shown in Figure 7. The application runs on the ARM core and handles I/O and application processing. To process the signals the application uses the VISA APIs which are provided by the Codec Engine along with a generic API called IUNIVERSAL. Codec Engine is divided into a CE Remote Server which runs on the DSP and a CE Client which runs on the ARM.

In order to fit in the Codec Engine framework, the recognition algorithms have to be compatible with xDAIS / xDM [10] standards form Texas Instruments, so that algorithm resource allocation is controlled by the application and that the application controls and uses the algorithms through a uniform set of APIs.

5. DSP IMPLEMENTATION

First the complete image recognition algorithm was implemented and tested in MATLAB as explained in section 3.2. This MATLAB implementation is used as reference throughout the DSP implementation on DM6446 using the EVMDM6446 from Spectrum Digital [11] as development board.

We have implemented the different blocks of the algorithm in C language using fixed point arithmetic. This was partly done under Code Composer Studio (CCS v4) from Texas Instruments along with Blackhawk USB-JTAG emulator (BH-USB-560BP) [12] as debugging tool. To implement the CDCC and DCTPM blocks, we used as much as possible the functions from the Image library (IMGLIB2) and the Vision library (VLIB) provided by Texas Instruments [13, 14] such as Canny edge detection, 8x8 block DCT calculation (IMG_fdct_8x8) and IMG_quantize for JPEG quantization.

The algorithm blocks to be implemented in the software framework are (see section 3) BRRP, CDCC, DCTPM and JPEG decoder. Each of the blocks was implemented as a separate Codec Engine "codec". The BRRP, CDCC and DCTPM codecs were implemented using the IUNIVERSAL xDM interface. The JPEG decoder which is provided by Texas Instruments is delivered as IIMGDEC1 xDM codec [15]. A single Codec Engine server configured to include these four codecs is made available to the application. We have used the Codec Engine GenCodecPkg wizard [16] to generate each codec package containing starter code and then we have added the C code of the algorithm blocks into it. The testing application for each codec using File I/O was written by modifying the universal_copy example delivered with the Codec Engine software.

5.1 Testing

We have put the file system of the board on a server with a fixed IP address and mounted it using NFS. The board, configured to boot from Flash NAND, was also connected to the network and was assigned a fixed IP address. This allowed the access from remote location to the file system via Samba or NFS and facilitated the remote login and execution on the board using SSH or Telnet. This setup allows sharing of the board between users and the possibility to perform extensive testing automatically from MATLAB: first the input files are written from MATLAB on the board filesystem, execution of the application on the board is controlled from MATLAB using SSH, and the output files are read by MATLAB from the board filesystem to analyze the results.

Using this development setup, we have extensively tested the codecs with images from the "clean database" and the "real condition database" (see section 3.1). By comparing the results from the DSP implementation with the results from the MATLAB implementation, we have verified the proper functioning of the DSP implementation, and that the losses due to the fixed-point arithmetic of the C64x+ do not affect significantly the performance.

6. CURRENT STATUS AND FUTURE WORK

We have completed the MATLAB implementation of the image recognition algorithm. Concerning the implementation on DSP, each of the blocks of the algorithm, namely, BRRP, CDCC, DCTPM and JPEG decoder have been separately implemented and tested. Ongoing and future work is in concatenating these blocks performing extensive testing, and measuring if the overall performance of the implemented algorithm is within the initial device specifications, especially the response time which should not exceed 0.5 second. However, given the measured performance of each of the blocks, we expect the overall performance to be well within these specifications. On the ARM side, we have implemented the device management, the barcode detection algorithm, the image acquisition and the audio acquisition and playback. A hardware prototype has been designed and delivered to the manufacturer. Once this prototype is ready, the complete firmware including application and DSP server will be tested on it.

7. CONCLUSIONS

We have presented the development and the implementation of an image recognition algorithm on DM6446 using the Codec Engine framework, outlining the steps to go from the algorithm design down to the DSP implementation. This implementation is part of the development of PictoBar, the next generation of Alternative and Augmentative Communication portable device for improved speech rehabilitation therapy.

By using Davinci technology, and by including the requirements of low complexity for portability throughout all stages of the development, we could successfully implement fast response image recognition task on a portable device.

8. ACKNOWLEDGEMENTS

This work was partly supported by the Swiss Federal Office for Professional Education and Technology (OPET) through the Innovation Promotion Agency (CTI) under the Grant CTI 8811.2 PFNM-NM ("PictoBar"project)[17]. We are grateful to all the people who have contributed to this work: Mr. Christian Vaucher, Mr. Michel Guinand, Mr. Yvan Magnin and Mr. Yves Mühlebach from FST; Mr. Mario Aellen and Mr. Timothée Carron from Gigatec S.A. and Mr. Pascal Bach who has implemented the CDCC algorithm during his practical semester work carried out at EPFL-ESPLAB.

REFERENCES

[1] www.fst.ch

[2] www.fst.ch/babar/

[3] www.ovt.com

[4] TMS320DM6446 Digital Media System-on-Chip - http://focus.ti.com/docs/prod/folders/print/tms320dm6446.html

[5] http://focus.ti.com/docs/prod/folders/print/tlv320aic26.html

[6] J. Bracamonte, M. Ansorge, F. Pellandini and P.-A. Farine, "Low complexity Image Matching in the compressed domain by using DCT-phase", *Proc. of 6th COST 276 workshop*, 88-93, 2004.

[7] www.mayer-johnson.com/education/symbols/pcs/

[8] http://focus.ti.com/docs/toolsw/folders/print/dspbios5.html

[9] TI *Codec Engine Application Developer Users Guide*, Texas Instruments, edition d, September 2007.

[10] TI *xDAIS-DM (Digital Media) User Guide*, Texas Instruments, edition b, January 2007.

[11] Spectrum Digital Inc. *DaVinci-DM644x Evaluation Module Technical Reference*, Spectrum Digital Inc., edition e, March 2007.

[12] Usb560-bp jtag emulator. - www.blackhawk-dsp.com/products/USB560BP.aspx.

[13] TI *TMS320C64x+ DSP Image/Video Processing Library (v2.0.1)*, Texas Instruments, edition a, May 2008.

[14] TI *Vision Library (VLIB) Application Programming Interface Reference Guide*, Texas Instruments, edition c, November 2009.

[15] http://focus.ti.com/docs/toolsw/folders/print/tmdjpegd.html

[16] http://processors.wiki.ti.com/index.php/Codec_Engine_GenCodecPkg_Wizard_FAQ

[17] *PictoBar: Portable augmentative and alternative communication device with pictogram recognition functionality for the speech handicapped.* - www.aramis.admin.ch