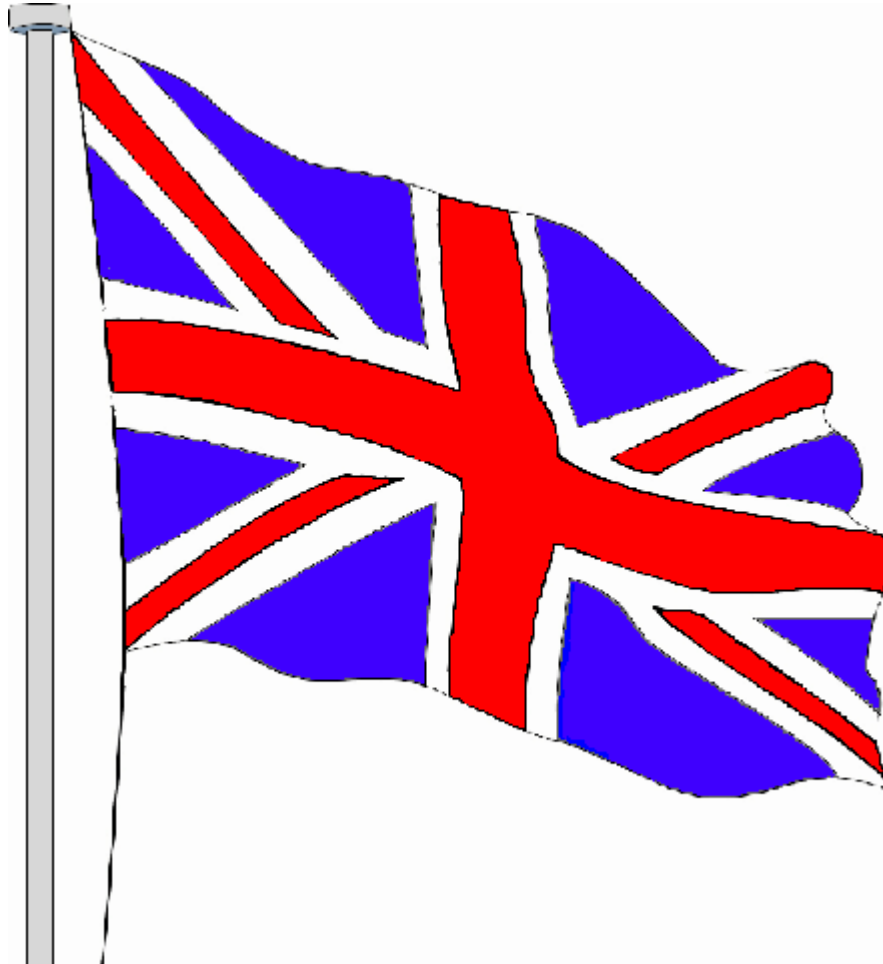




**-Preliminary-
ePanel-mgx
Technical Manual
Multimedia-Interface
Version 1.03**

English / Deutsch



English version

1. INTRODUCTION

The current version of the Multimedia Interface supports **Video-Input only**. Audio- or Video-Output functionality is supported by the BIOS or by the operating system drivers, i.e. the combination Video-Input to Video-Output is not implemented. The video input stream can be displayed on a standard Monitor, a Flatpanel or both simultaneously. Supported video input signals are: Composite, S-Video and SCART (RGB).

The interface address is located in the I/O address range and can be accessed by any operating system. The base address is 0x0A40 and contains an index and a data register, each 16 bits wide. The interface allows direct access to the video decoder chip SAA7118 (PHILIPS) and to the video signal related registers of the Geode processor SC1200 (NATIONAL/AMD).

Supported display resolutions are SVGA and XGA. Use of a VGA panel is not recommended. Any possible color depth is supported. When scaling video images, the combination of Up- and Down-scaling from basis resolution (720x576 for PAL and 720x480 for NTSC) leads to unsatisfactory results and is not recommended (e.g. 800x400 or 700x600 pixel). Examples of valid resolutions are 200x150, 400x300, 800x600 or 1024x768 pixel. Changing the aspect ratio is also supported, for example 100x400 pixel.

The Multimedia interface supports color keying and alpha blending. Alpha blending allows up to three different windows. To use alpha blending, **color keying has to be enabled first**.

A freeze function allows easy capturing of images from the input video stream. After freezing applications can easily create BMP-files from the image.

2. FUNCTION OVERVIEW

Basic functions:

<i>Videoln_Init</i>	initialising and displaying the video image
<i>Videoln_HPos</i>	horizontal position of the video image
<i>Videoln_VPos</i>	vertical position of the video image
<i>Videoln_HSize</i>	video image width
<i>Videoln_VSize</i>	video image height
<i>Videoln_Scale</i>	alternative method of setting width/height by specifying a scale ratio (example: 3/5 of full size)
<i>Videoln_Contrast</i>	video contrast
<i>Videoln_Brightness</i>	video brightness
<i>Videoln_Hue</i>	color hue
<i>Videoln_Saturation</i>	color saturation
<i>Videoln_Command</i>	special functions (freeze image, move position, change size, start capturing, stop video display)

Color-key functions (optional):

<i>Videoln_Color</i>	color-key color
<i>Videoln_Mask</i>	color-key mask (mask color areas)
<i>Videoln_Special</i>	special functions (select video or graphics as color-keying source)

Alpha-blending functions (optional):

<i>Videoln_Num_Prior</i>	current window number and priority
<i>Videoln_Alpha_HPos</i>	horizontal window position
<i>Videoln_Alpha_VPos</i>	vertical window position
<i>Videoln_Alpha_HSize</i>	window width
<i>Videoln_Alpha_VSize</i>	window height
<i>Videoln_Alpha_Value</i>	alpha-blending value
<i>Videoln_Alpha_Color</i>	window color
<i>Videoln_Special</i>	special functions (activate alpha-blending and move window)

Video image positioning functions (optional):

<i>Videoln_Pos_HPos</i>	move horizontal position in the decoder chip (video input)
<i>Videoln_Pos_VPos</i>	move vertical position in the decoder chip (video input)

Capture functions (optional):

<i>Videoln_YUV_Ptr</i>	pointer for copying the image
<i>Videoln_YUV_IO</i>	read the video data via I/O addresses

Register functions (optional):

<i>Videoln_SAA7118</i>	access register of video decoder SAA7118
<i>Videoln_GEODE</i>	access register of processor Geode SC1200

Status / Error functions:

<i>Videoln_Status</i>	status information
<i>Videoln_Errorcode</i>	error code

3. INTERFACE DESCRIPTION

Index register address: **0x0A40**
 Data register address: **0x0A42**

Basic functions

Videoln Init

Initialize and show video image.

Index: **0x0000**

Data: Bit 0

0 = NTSC 1 = PAL

Bit 1

0 = standard channel selection 1 = extended channel selection
 (currently only standard selection has been implemented)

Bit 2

0 = Composite 1 = S-Video

Bit 4

0 = Input channel 1 1 = Input channel 2

Bit 8

0 = BOB-Deinterlacing 1 = WEAVE-Deinterlacing

Notes:

This function is intended to be used after all other parameters have been set. At a minimum, position and size of the video image should have been set. With the exception of Bit 8 the control Bits don't need any further explanation. BOB uses only half of the video input image, reducing the image quality, but requiring only half the bandwidth and optimizing the processor power. WEAVE displays the full image, but unbuffered (single buffering).

Videoln HPos

Horizontal position of the video image

Index: **0x0001**

Data: **variable**

valid range: 0x0000 – 0x03FF

Videoln VPos

Vertical position of the video image

Index: **0x0002**

Data: **variable**

valid range: 0x0000 – 0x02FF

Videoln HSize

Horizontal size of the video image

Index: **0x0003**

Data: **variable**

valid range: 0x0032 – 0x0400

Notes:

Always set both, HSize and VSize. Failing to do so will result in an error message.

Videoln VSize

Vertical size of the video image

Index: **0x0004**
Data: **variable**
valid range: 0x0028 – 0x0300

Videoln Scale

Alternative dimensional information obtained on the standard format PAL or NTSC (replacing HSize and VSize)

Index: **0x0005**
Data: **LO-Byte - divisor (variable)**
HI-Byte - dividend (variable)
valid range: 0x01 – 0xFF

Notes:

The divisor must always be larger than or equal to the dividend. The resolution of the video standard (e.g. PAL = 720x576 pixel) serves as a base. Examples: Setting the register to 0x0102 reduces the image size to one quarter of the original image, setting the register to 0x0101 creates full size.

Videoln Contrast

Adjust contrast

Index: **0x0006**
Data: **variable**
valid range: 0x0000 – 0x00FF

Videoln Brightness

Adjust brightness

Index: **0x0007**
Data: **variable**
valid range: 0x0000 – 0x00FF

Videoln Hue

Adjust hue

Index: **0x0008**
Data: **variable**
valid range: 0x0000 – 0x00FF

Videoln Saturation

Adjust saturation

Index: **0x0009**
Data: **variable**
valid range: 0x0000 – 0x00FF

Videoln Command

Special functions (the function will be executed once)

Index: **0x000A**
Data: **Bit 0**
Freeze video image
Bit 1
Thaw video image (normal mode)

Bit 2

Move video image to a new position

Bit 3

Enlarge or reduce the video image

Bit 4

Start capturing the video image

Bit 15

Stop video display

Notes:

Bits 0 and 1 don't need further explanation. Before setting Bits 2, 3 or 4, you must set the corresponding parameters for the function (Bit 2 requires HPos/VPos, Bit 3 requires HSize/VSize, Bit 4 requires a valid memory pointer). Bit 15 clears the video display, to restart the display again use VideoIn_Init. Note that VideoIn_Init resets all parameters to their default values.

Color-key functions

Videoln Color

Set the color-key color

Index: **0x0010 and 0x0011**

Data: **variable**

valid range: 0x00000000 – 0x00F8FCF8

Notes:

Color values are always represented in 5-6-5 scheme. Therefore the maximum value is coded as 0x00F8FCF8 (white) and not as 0x00FFFFFF. As the color value contains 24 Bit – more than one index is necessary. Consequently the index 0x0010 contain the GB-part (LO-Byte = Blue, HI-Byte = Green) and index 0x0011 the R-part (LO-Byte = Red).

Videoln Mask

Set the color-key mask

Index: **0x0012 and 0x0013**

Data: **variable**

valid range: 0x00000000 – 0x00F8FCF8

Notes:

See Videoln_Color

Alpha-blending functions

Videoln Num Prior

Define alpha-blending number and priority

Index: **0x0020**

Data: **LO-Byte - number (variable)**

HI-Byte - priority (variable)

valid range for both bytes: 0x00 – 0x02

Notes:

The processor can handle up to three alpha-blending windows. The windows **must** have different priority values, where 0 is the lowest, 2 the highest priority.

Videoln Alpha HPos

Horizontal alpha-blending window position

Index: **0x0021**

Data: **variable**

valid range: 0x0000 – 0x03FF

Notes:

The horizontal position must lie within the actual video image.

Videoln Alpha VPos

Vertical alpha-blending window position

Index: **0x0022**

Data: **variable**

valid range: 0x0000 – 0x02FF

Notes:

The vertical position must lie within the actual video image.

Videoln Alpha HSize

Width of the alpha-blending window

Index: **0x0023**

Data: **variable**

valid range: 0x0001 – 0x0400

Notes:

The alpha-blending window must reside fully within the video image. Trying to set an alpha-blending window size and position so that parts of the alpha-blending window lie outside the video image will result in an error message.

Videoln Alpha VSize

Height of the alpha-blending window

Index: **0x0024**

Data: **variable**

valid range: 0x0001 – 0x0300

Notes:

See Videoln_Alpha_HSize

Videoln Alpha Value

Set the alpha-blending value

Index: **0x0025**

Data: **variable**

valid range: 0x0000 – 0x00FF

Notes:

Setting the alpha-blending value to 0 causes the video image to be shown, setting the value to 0xFF shows only the graphical background. All other values blend video and background image together.

Videoln Alpha Color

Define alpha-blending color

Index: **0x0026 and 0x0027**

Data: **variable**

valid range: 0x00000000 – 0x00F8FCF8

Notes:

Color values are always represented in 5-6-5 scheme. Therefore the maximum value is coded as 0x00F8FCF8 (white) and not as 0x00FFFFFF. As the color value contains 24 Bit – more than one index is necessary. Consequently the index 0x0026 contain the GB-part (LO-Byte = Blue, HI-Byte = Green) and index 0x0027 the R-part (LO-Byte = Red). Setting a value of 0 disables the function.

Videoln Special

Special functions (the function will be executed once)

Index: **0x0030**

Data: **Bit 0**

activate color-keying for the video image

Bit 1

activate color-keying for the graphics background

Bit 2

deactivate color-keying

Bit 8

move alpha-blending window

Bit 9

activate alpha-blending window for graphics background

Bit 10

activate alpha-blending window for video image

Bit 11

deactivate alpha-blending

Notes:

Color-keying for the video image means: all pixels in the video image matching the color-keying value are replaced with the background image. Color-keying for the graphics background works in the same way. The color-keying value must have been set before activating this function.

First, before setting any other parameters for alpha-blending, define the window number and priority. The alpha-blending positioning needs horizontal/vertical window position and horizontal/vertical window size. The window position can be set before and after alpha-blending has been activated. However the activation needs alpha-blending value and color (usually zero). Bit 9 and 10 determine the image that is shown outside the alpha-blending window: video image or background graphics.

IMPORTANT: Alpha-blending requires color-keying! If color-keying has not been activated, alpha-blending will show no effect.

NOTE:

Reading the data register returns the values that have been set. Write the index number into the index register first, then read the data register to read a value.

Video image positioning functions

VideoIn Pos HPos

Move video image in the decoder (video input) horizontally

Index: **0x0040**

Data: **variable**

valid range: 0x0000 – 0x0FFF

VideoIn Pos VPos

Move video image in the decoder (video input) vertically

Index: **0x0041**

Data: **variable**

valid range: 0x0000 – 0x0FFF

Notes:

Playback of video sources (e.g. DVD) frequently contains interference patterns in the first few scan lines. This function can remove these lines from the video image.

Capture functions

Videoln YUV Ptr

Define the pointer for capturing. Use **Videoln_Command** to capture the video image.

LO-Word pointer:

Index: **0x0050**
Data: **variable**
valid range: 0x0000 – 0xFFFF

HI-Word pointer:

Index: **0x0051**
Data: **variable**
valid range: 0x0001 – 0xFFFF

Notes:

The interface copies the video data to the memory location defined by the pointer. The maximum amount of data transferred is 720 x 576 x 2 Bytes. The pointer must contain a valid physical memory address. If you are using an operating system with memory management such as Windows or Linux, you must call the appropriate kernel function to convert a virtual to the physical address. Note also that the physical address is always a linear address, not an segment/offset combination. Passing a NULL pointer is not allowed. Please see Appendix B for further information about the data format. If freezing not already active, this function will be set to 'freeze' automatically.

Videoln YUV IO

Reset data pointer (write access) and get video data (read access)

Write access:

Reset the data pointer

Index: **0x0052**
Data: **variable**
valid range: 0x0000 – 0xFFFF

Read access:

Get video data

Index: **0x0052**
Data: **variable**

Notes:

The alternative method for capturing a video image uses I/O-commands to read the video data. Use this method when you can't get a physical memory address from the operating system. To capture the video image, first write to the data register at index 0x0052. This sets the data pointer to 0. Then read the data register continually until you have captured the entire video image. Capturing a PAL video image at maximum size requires 720 x 576 reads. Smaller video images require fewer reads (e.g. video resolution of 200x150 requires 200 x 150 reads). Larger video images do not increase the number of reads. If you have enlarged the video image to a resolution of 800x600, a PAL video image will still require only 720 x 576 consecutive reads.

IMPORTANT:

If you want to capture a video image, the horizontal size **must** be a multiple of 8 and the vertical size **must** be a multiple of 2. Other resolutions can cause garbled images.

Register functions

Videoln SAA7118

Read and write registers of the SAA7118 video decoder (PHILIPS)

Write access

Writing data

Index: **0x0100**
Data: **LO-Byte - register (variable)**
HI-Byte - data (variable)
valid range for both Bytes: 0x00 – 0xFF

Register definition for read access

Index: **0x101**
Data: **LO-Byte - register (variable)**
valid range: 0x00 – 0xFF

Read access

Reading a register

Index: **0x101**
Data: **LO-Byte - data (variable)**

Notes:

Every read access requires a renewed register definition (write access).
Please consult the SAA7118 documentation for further details about the registers.

Videoln GEODE

Read and write registers of the SC1200 Geode processor (NATIONAL/AMD)

Write access

Writing data

Register definiton

Index: **0x0102**
Data: **LO-Byte - register (variable)**
HI-Byte (lower nibble) - register (variable)
Bit 15
0 = F4BAR0 1 = F4BAR2
valid range: 0x0000 – 0x0FFF

LO-Word data

Index: **0x0103**
Data: **variable**
valid range: 0x0000 – 0xFFFF

HI-Word data

Index: **0x0104**
Data: **variable**
valid range: 0x0000 – 0xFFFF

Register definition for read access

Index: **0x0105**
Data: **LO-Byte - register (variable)**
HI-Byte (lower nibble) - register (variable)
Bit 15
0 = F4BAR0 1 = F4BAR2
valid range: 0x0000 – 0x0FFF

LO-Word data

Index: **0x0103**
Data: **variable**

HI-Word data

Index: **0x0104**
Data: **variable**

Notes:

Every read access requires a renewed register definition (write access). The order for writing to a Geode processor register is: LO-Word data, HI-Word data and then register offset.

Please consult the Geode SC1200 reference manual for further details about processor registers.

Status / Error Functions

Videoin Status

Read interface status

Index: **0x0200**

Data: **Bit 0**

0 = Video not initialised 1 = Video initialised

Bit 14

0 = No error 1 = Error occurred

Bit 15

0 = Ready to accept data 1 = Not ready

Notes:

Always read the error code (see Videoin_Errorcode) when Videoin_Status flags an error. Some video functions will not be executed unless the error code has been read.

Videoin Errorcode

Read error code

Index: **0x201**

Data: **variable**

Error codes:

VIDEOIN_ERROR_NOERR	0x0000
VIDEOIN_ERROR_HPOS	0x0001
VIDEOIN_ERROR_VPOS	0x0002
VIDEOIN_ERROR_HSIZE	0x0003
VIDEOIN_ERROR_VSIZE	0x0004
VIDEOIN_ERROR_SCALE	0x0005
VIDEOIN_ERROR_PTR	0x0020
VIDEOIN_ERROR_ALPHA_NUM	0x0100
VIDEOIN_ERROR_ALPHA_PRIOR	0x0101
VIDEOIN_ERROR_ALPHA_HPOS	0x0102
VIDEOIN_ERROR_ALPHA_VPOS	0x0103
VIDEOIN_ERROR_ALPHA_HSIZE	0x0104
VIDEOIN_ERROR_ALPHA_VSIZE	0x0105

Notes:

The names of the error codes are self-explanatory. Errors usually indicate that a value has been passed to a function that is outside the valid range. Some functions will not cause an error, parameters will not be validated and an error code will not be set until the corresponding command is executed.

Appendix A

Examples:

Example 1

Minimal initialisation

```

outpw (0x0A40, 0x0001);
outpw (0x0A42, 0x0014);           // horiz. pos. = 20
outpw (0x0A40, 0x0002);
outpw (0x0A42, 0x000A);         // vert. pos. = 10
outpw (0x0A40, 0x0003);
outpw (0x0A42, 0x01F4);         // horiz. size = 500
outpw (0x0A40, 0x0004);
outpw (0x0A42, 0x0190);         // vert. size = 400
outpw (0x0A40, 0x0000);
outpw (0x0A42, 0x0011);         // Initialise
                                   // PAL, Composite
                                   // Input channel 2
                                   // BOB-Deinterlacing

```

Example 2

Accessing GEODE processor register to freeze the video image

```

VOID SetFreeze (BOOL bFreeze)
{
    USHORT nDataLO, nDataHI;
    outpw (0x0A40, 0x0105);       // read register
    outpw (0x0A42, 0x8004);       // VIP_CONTROL in F4BAR2
    outpw (0x0A40, 0x0103);
    nDataLO = inpw (0x0A42);
    outpw (0x0A40, 0x0104);
    nDataHI = inpw (0x0A42);

    nDataLO &= 0xFFFC;
    if (bFreeze)
        nDataLO |= 0x0001;       // freeze
    else
        nDataLO |= 0x0003;       // unfreeze

    outpw (0x0A40, 0x0103);
    outpw (0x0A42, nDataLO);
    outpw (0x0A40, 0x0104);
    outpw (0x0A42, nDataHI);
    outpw (0x0A40, 0x0102);
    outpw (0x0A42, 0x8004);       // write register
}

```

Appendix B

Structure of video data (capture functions):

Video data be present in YUV-Format (4:2:2), i.e. one double word (32 bit) contains the color information of two pixels. A double word appears in the following manner (each one byte): **Y** (first pixel) - **U** - **Y** (second pixel) - **V**.

The maximum amount of memory for a PAL image is 720 x 576 words (= 829440 bytes), the maximum memory needed for an NTSC image is 720 x 480 words (= 691200 bytes). The start address is constant, the video window position has no effect. Resolutions lower than the standard resolutions require less memory, e.g. 400 x 300 pixels require 400 x 300 words (= 240000 bytes).

Physical start address of video memory = 0x409C0000.

You can use this address to access the video data directly from memory as an alternative to using VideoIn_YUV_Ptr and VideoIn_YUV_IO.

Use the following calculation to convert YUV to 24-Bit RGB format.

$$\begin{aligned} R &= 1.1640625 (Y - 16) + 1.59375 (V - 128) \\ G &= 1.1640625 (Y - 16) - 0.8125 (V - 128) - 0.390625 (U - 128) \\ B &= 1.1640625 (Y - 16) + 2.015625 (U - 128) \end{aligned}$$

Black (RGB 0,0,0) has the YUV-value 0x10801080. The calculations can return negative values or values greater than 255. It is recommended to clip the results of the YUV - RGB calculations to a range of 0 - 255.



Deutsche Version

1. EINLEITUNG

Die aktuelle Version umfaßt derzeit **nur Funktionen für Video-Input**. Audio- bzw. Video-Output-Unterstützung bieten allein das BIOS oder die verfügbaren Betriebssystem-Treiber, d.h. auch die Kombinationen aus Video-Input und Video-Output ist nicht Bestandteil des Interfaces. Die Darstellung der Video-Daten kann entweder auf einem handelsüblichen Monitor oder einem LCD-Bildschirm oder beiden zugleich erfolgen. Als Video-Eingangssignale kommen in Betracht: Composite, S-Video und Scart (RGB).

Die Schnittstelle selbst liegt im I/O-Bereich und läßt sich somit aus jedem Betriebssystem ansprechen. Die Basisadresse lautet 0x0A40 und umfaßt ein Index- und ein Datenregister (jeweils 16 Bit breit). Das Interface bietet auch einen direkten Zugriff auf den Video-Decoder SAA7118 (PHILIPS) und den maßgeblichen Video-Registern des Prozessors Geode SC1200 (NATIONAL/AMD).

Bei der Bildschirmauflösung unterstützt die Schnittstelle derzeit SVGA- und XGA-Auflösung. Der Einsatz eines VGA-Panels ist daher nicht sinnvoll. Die eingestellte Farbtiefe spielt dabei keine Rolle. Bezogen auf die Basisauflösungen 720x576 Pixel (PAL) bzw. 720x480 Pixel (NTSC) verbietet sich allerdings eine Kombination aus Up- und Down-Scaling (z.B. 800x400 oder 700x600 Pixel). Erlaubt sind hingegen Bildauflösungen wie 200x150, 400x300, 800x600, 1024x768 oder Verzerrungen wie etwa 100x400 Pixel.

Optional lassen sich Color-Key und Alpha-Blending aktivieren. Alpha-Blending bietet maximal drei unabhängige Fenster. **Ohne Einschalten der Color-Key Funktion** läßt sich Alpha-Blending allerdings nicht nutzen.

Ein weiteres Highlight stellt die Capture-Funktion dar. Nach dem notwendigen Einfrieren des Bildes kann beispielsweise mühelos eine BMP-Datei erzeugt werden.

2. FUNKTIONSÜBERBLICK

Grundfunktionen:

<i>Videoln_Init</i>	Initialisierung und Anzeige des Video-Bildes
<i>Videoln_HPos</i>	Horizontale Position des Video-Bildes
<i>Videoln_VPos</i>	Vertikale Position des Video-Bildes
<i>Videoln_HSize</i>	Horizontale Größe des Video-Bildes
<i>Videoln_VSize</i>	Vertikale Größe des Video-Bildes
<i>Videoln_Scale</i>	Alternativ zu HSize/VSize Angabe als Dividend/Divisor (z.B. 3/5 Vollbild)
<i>Videoln_Contrast</i>	Kontrasteinstellung
<i>Videoln_Brightness</i>	Helligkeitseinstellung
<i>Videoln_Hue</i>	Farbtoneinstellung
<i>Videoln_Saturation</i>	Farbsättigungseinstellung
<i>Videoln_Command</i>	Spezielle Funktionen (Bild einfrieren, Bildposition ändern, Bildgröße ändern, Capturing starten und Video-Anzeige beenden)

Color-Key-Funktionen (optional)

<i>Videoln_Color</i>	Color-Key Farbe
<i>Videoln_Mask</i>	Color-Key Maske (ausmaskieren von Farbbereichen)
<i>Videoln_Special</i>	Spezielle Funktionen (Color-Key für Video oder Color-Key für unterlegte Grafik aktivieren)

Alpha-Blending-Funktionen (optional)

<i>Videoln_Num_Prior</i>	Aktuelle Fensternummer und Fensterpriorität
<i>Videoln_Alpha_HPos</i>	Horizontale Position des Fensters
<i>Videoln_Alpha_VPos</i>	Vertikale Position des Fensters
<i>Videoln_Alpha_HSize</i>	Horizontale Größe des Fensters
<i>Videoln_Alpha_VSize</i>	Vertikale Größe des Fensters
<i>Videoln_Alpha_Value</i>	Blendwert
<i>Videoln_Alpha_Color</i>	Farbe des Fensters
<i>Videoln_Special</i>	Spezielle Funktionen (Alpha-Blending einschalten und Fenster verschieben)

Bildpositionierungs-Funktionen (optional)

<i>Videoln_Pos_HPos</i>	Videobild im Decoder (Video-Input) horizontal verschieben
<i>Videoln_Pos_VPos</i>	Videobild im Decoder (Video-Input) vertikal verschieben

Capture-Funktionen (optional)

<i>Videoln_YUV_Ptr</i>	Zeiger für das Kopieren eines Bildes
<i>Videoln_YUV_IO</i>	Bildinformationen über I/O-Adresse auslesen

Registerfunktionen (optional)

<i>Videoln_SAA7118</i>	Registerzugriff auf Video-Decoder SAA7118
<i>Videoln_GEODE</i>	Registerzugriff auf Prozessor Geode SC1200

Status-/Fehlerfunktionen

<i>Videoln_Status</i>	Statusinformationen
<i>Videoln_Errorcode</i>	Fehlercode

3. SCHNITTSTELLENFUNKTIONEN

Adresse Index-Register: **0x0A40**
 Adresse Daten-Register: **0x0A42**

Grundfunktionen

Videoln Init

Initialisierung und Anzeige des Video-Bildes

Index: **0x0000**
 Daten: **Bit 0**
 0 = NTSC 1 = PAL
Bit 1
 0 = Standard-Kanalwahl 1 = Extended-Kanalwahl
 (derzeit nur Standard-Kanalwahl implementiert)
Bit 2
 0 = Composite 1 = S-Video
Bit 4
 0 = Eingangskanal 1 1 = Eingangskanal 2
Bit 8
 0 = BOB-Deinterlacing 1 = WEAVE-Deinterlacing

Anmerkungen:

Diese Funktion sollte erst ganz am Schluß einer Parameterübergabe zur Ausführung kommen. Zumindest die Definition von Position und Größe des Video-Bildes sind Voraussetzungen für die Aktivierung dieses Kommandos. Mit Ausnahme von Bit 8 bedürfen die Steuerbits wohl keiner weiteren Erläuterung. Das BOB-Deinterlacing benutzt nur ein Halbbild für die Darstellung und weist damit zumindest theoretisch die schlechtere Bildqualität auf. Beim WEAVE-Deinterlacing erfolgt die Darstellung eines Vollbildes, allerdings ungepuffert (Single-Buffering). Der Vorteil bei BOB liegt in der geringeren benötigten Bandbreite bei der Datenübertragung, d.h. die Prozessorbelastung ist geringer.

Videoln HPos

Horizontale Position des Video-Bildes

Index: **0x0001**
 Daten: **variabel**
 Zulässiger Bereich: 0x0000 - 0x03FF

Videoln VPos

Vertikale Position des Video-Bildes

Index: **0x0002**
 Daten: **variabel**
 Zulässiger Bereich: 0x0000 – 0x02FF

Videoln HSize

Horizontale Größe des Video-Bildes

Index: **0x0003**
 Daten: **variabel**
 Zulässiger Bereich: 0x0032 – 0x0400

Anmerkungen:

Die Übergabe nur eines Parameters (z.B. nur HSize ohne VSize) führt zu einer Fehlermeldung.

Videoln VSize

Vertikale Größe des Video-Bildes

Index: **0x0004**

Daten: **variabel**

Zulässiger Bereich: 0x0028 – 0x0300

Videoln Scale

Alternative Größenangabe bezogen aufs Standardformat 720x576 / 720x480 (ersetzt HSize / VSize)

Index: **0x0005**

Daten: **LO-Byte - Divisor (variabel)**

HI-Byte - Dividend (variabel)

Zulässiger Bereich für beide Bytes: 0x01 – 0xFF

Anmerkungen:

Der Divisor muß immer größer/gleich dem Dividend sein. Als Basis dient die Auflösung des Standardbildes (z.B. 720x576 Pixel bei PAL). Die Übergabe von 0x0102 würde das Bild horizontal und vertikal halbieren, der Wert 0x0101 generiert ein Vollbild.

Videoln Contrast

Kontrasteinstellung

Index: **0x0006**

Daten: **variabel**

Zulässiger Bereich: 0x0000 – 0x00FF

Videoln Brightness

Helligkeitseinstellung

Index: **0x0007**

Daten: **variabel**

Zulässiger Bereich: 0x0000 – 0x00FF

Videoln Hue

Farbtoneinstellung

Index: **0x0008**

Daten: **variabel**

Zulässiger Bereich: 0x0000 – 0x00FF

Videoln Saturation

Farbsättigungseinstellung

Index: **0x0009**

Daten: **variabel**

Zulässiger Bereich: 0x0000 – 0x00FF

Videoln Command

Spezielle Funktionen (einmalige Ausführung nur bei gesetztem Bit)

Index: **0x000A**

Daten: **Bit 0**

Video-Bild einfrieren

Bit 1

Video-Bild aus eingefrorenem Zustand lösen (Normalmodus)

Bit 2

Video-Bild auf neue Position bewegen

Bit 3

Video-Bild vergrößern oder verkleinern

Bit 4

Capturing des Video-Bildes starten

Bit 15

Video-Anzeige beenden

Anmerkungen:

Bit 0 und Bit 1 benötigen keine Erklärung. **Vor** der Aktivierung von Bit 2 sollte eine Übergabe von neuen Positionsdaten (HPos und VPos) stattgefunden haben, für Bit 3 gilt Gleiches für die Grössenangaben (HSize und VSize) und für Bit 4 bezüglich des Zeigers. Bit 15 löscht die Video-Darstellung, ein Neustart ist nur mit der Init-Funktion möglich. Alle Parameter werden dabei auf ihren Defaultwert zurückgesetzt.

Color-Key Funktionen

VideIn Color

Color-Key Farbe definieren

Index: **0x0010 und 0x0011**

Daten: **variabel**

Zulässiger Bereich: 0x00000000 – 0x00F8FCF8

Anmerkungen:

Die Übergabe von Farbwerten erfolgt generell im 5-6-5 Schema. Daher beträgt der Maximalwert nicht 0x00FFFFFF sondern 0x00F8FCF8 (Farbe Weiß). Der nicht komprimierte Farbwert umfaßt also 24 Bits und kann daher nicht mit einem Index bewältigt werden. Folglich beinhaltet der Index 0x0010 den GB-Teil (LO-Byte = Blauteil, HI-Byte = Grünteil) und der Index 0x0011 den R-Teil (LO-Byte).

VideIn Mask

Color-Key Maske definieren

Index: **0x0012 und 0x0013**

Daten: **variabel**

Zulässiger Bereich: 0x00000000 – 0x00F8FCF8

Anmerkungen:

Es gelten die gleichen Regeln wie bei VideIn_Color.

Alpha-Blending Funktionen

Videoln Num Prior

Alpha-Blending Nummer und Priorität definieren

Index: **0x0020**

Daten: **LO-Byte - Nummer (variabel)**

HI-Byte - Priorität (variabel)

Zulässiger Bereich für beide Bytes: 0x00 – 0x02

Anmerkungen:

Der Prozessor vermag bis zu drei Alpha-Blending Fenster zu verwalten. Die Fenster lassen sich mit einer Priorität versehen (0 entspricht der niedrigsten, 2 der höchsten Priorität). Zwei Fenster dürfen niemals die gleiche Priorität besitzen.

Videoln Alpha HPos

Alpha-Blending horizontale Fenster-Position

Index: **0x0021**

Daten: **variabel**

Zulässiger Bereich: 0x0000 – 0x03FF

Anmerkungen:

Die horizontale Position des Alpha-Blending Fensters darf nicht außerhalb des eigentlichen Video-Bildes zu liegen kommen.

Videoln Alpha VPos

Alpha-Blending vertikale Fenster-Position

Index: **0x0022**

Daten: **variabel**

Zulässiger Bereich: 0x0000 – 0x02FF

Anmerkungen:

Die vertikale Position des Alpha-Blending Fensters darf nicht außerhalb des eigentlichen Video-Bildes zu liegen kommen.

Videoln Alpha HSize

Alpha-Blending horizontale Fenster-Größe

Index: **0x0023**

Daten: **variabel**

Zulässiger Bereich: 0x0001 – 0x0400

Anmerkungen:

Das Alpha-Blending Fenster darf nicht größer werden als das eigentliche Video-Bild. Das Fenster muß gänzlich im Video-Bild enthalten sein. Eine nur teilweise Überlappung führt zu einer Fehlermeldung.

Videoln Alpha VSize

Alpha-Blending vertikale Fenster-Größe

Index: **0x0024**

Daten: **variabel**

Zulässiger Bereich: 0x0001 – 0x0300

Anmerkungen:

Das Alpha-Blending Fenster darf nicht größer werden als das eigentliche Video-Bild. Das Fenster muß gänzlich im Video-Bild enthalten sein. Eine nur teilweise Überlappung führt zu einer Fehlermeldung.

Videoln Alpha Value

Alpha-Blending Wert definieren

Index: **0x0025**

Daten: **variabel**

Zulässiger Bereich: 0x0000 – 0x00FF

Anmerkungen:

Die Nullstellung läßt nur das Video-Bild im Vordergrund erscheinen, die Maximalstellung nur den grafischen Hintergrund. Dazwischen erfolgt ein Mischen von Video-Bild und Hintergrundgrafik.

Videoln Alpha Color

Alpha-Blending Farbe definieren

Index: **0x0026 und 0x0027**

Daten: **variabel**

Zulässiger Bereich: 0x00000000 – 0x00F8FCF8

Anmerkungen:

Die Übergabe von Farbwerten erfolgt generell im 5-6-5 Schema. Daher beträgt der Maximalwert nicht 0x00FFFFFF sondern 0x00F8FCF8 (Farbe Weiß). Der nicht komprimierte Farbwert umfaßt also 24 Bits und kann daher nicht mit einem Index bewältigt werden. Folglich beinhaltet der Index 0x0026 den GB-Teil (LO-Byte = Blauteil, HI-Byte = Grünteil) und der Index 0x0027 den R-Teil (LO-Byte). Der Wert Null markiert das Abschalten dieser Funktion, alle anderen Farbwerte überlagern im definierten Fenster den Bildinhalt.

Videoln Special

Spezielle Funktionen (einmalige Ausführung nur bei gesetztem Bit)

Index: **0x0030**

Daten: **Bit 0**

Color-Key für das Video-Bild aktivieren

Bit 1

Color-Key für den Grafik-Hintergrund aktivieren

Bit 2

Color-Key deaktivieren

Bit 8

Alpha-Blending Fenster verschieben

Bit 9

Alpha-Blending Fenster mit Grafik-Hintergrund aktivieren

Bit 10

Alpha-Blending Fenster mit Video-Bild aktivieren

Bit 11

Alpha-Blending deaktivieren

Anmerkungen:

Color-Key für das Video-Bild bedeutet, daß bei Übereinstimmung mit der gewählten Farbe im Video-Bild die Hintergrund-Grafik erscheint. Analog verhält sich die Einstellung Color-Key für Grafik-Hintergrund. Die Color-Key Farbe muß selbstverständlich **vor** der Aktivierung definiert worden sein.

Die Alpha-Blending Funktionalität gliedert sich in zwei Teile: Positionsübergabe und eigentliche Aktivierung. Es empfiehlt sich, vor dem Einschalten die Position festzulegen. Sowohl für die Position als auch für die Aktivierung besteht die Notwendigkeit, die Fensternummer und Fensterpriorität **vorher** festzulegen. Die Positionsübergabe erfordert zusätzlich horizontale und vertikale Fenster-Position sowie horizontale und vertikale Fenster-Größe. Eine Positionsänderung kann jederzeit nach Aktivierung des Fensters erfolgen. Die Aktivierung hingegen benötigt neben Fensternummer/-priorität den Blendwert und die Farbe (letztere sollte gewöhnlich Null enthalten). Bit 9 und Bit 10 legen zusätzlich fest, womit der Bereich **außerhalb** des Fensters gefüllt wird: Video-Bild oder Hintergrund-Grafik.

WICHTIG: Ohne Aktivierung der Color-Key Funktion zeigt Alpha-Blending keine Wirkung.

HINWEIS:

Die oben aufgeführte Funktionserläuterung bezieht sich auf Schreibvorgänge. Ein Lesebefehl für die entsprechenden Indizes führt zu einem Rücklesen der übergebenen Werte.

Bildpositionierungs-Funktionen

Videoln Pos HPos

Videobild im Decoder (Video Input) horizontal verschieben

Index: **0x0040**

Daten: **variabel**

Zulässiger Bereich: 0x0000 – 0x0FFF

Videoln Pos VPos

Videobild im Decoder (Video Input) vertikal verschieben

Index: **0x0041**

Daten: **variabel**

Zulässiger Bereich: 0x0000 – 0x0FFF

Anmerkungen:

Beim Abspielen z.B. von DVDs gibt es häufig einen störenden Streifen am oberen Rand, der mit dieser Funktion entfernt werden kann.

Capture-Funktionen

Videoln YUV Ptr

Zeiger für das Capturing übergeben. Der Start der Capture-Funktion erfolgt mit **Videoln_Command**.
LO-WORD Zeiger

Index: **0x0050**
Daten: **variabel**
Zulässiger Bereich: 0x0000 – 0xFFFF

HI-WORD Zeiger

Index: **0x0051**
Daten: **variabel**
Zulässiger Bereich: 0x0001 – 0xFFFF

Anmerkungen:

Die Schnittstelle kopiert Videodaten in den Bereich, der durch den Zeiger definiert wird. Der maximale Umfang der übertragenen Daten liegt bei 720 x 576 x 2 Bytes. Der Zeiger darf natürlich nur auf physikalisch vorhandenen Speicher verweisen, nicht wie z.B. bei WINDOWS üblich auf virtuelle Bereiche, und beinhaltet eine lineare Adresse (also nicht Segment/Offset). Die Übergabe eines NULL-Zeigers ist nicht erlaubt. Weitere Details bezüglich der Datenstruktur siehe Anhang B. Das Einfrieren des Bildes erfolgt automatisch, sofern dies nicht bereits im Vorfeld geschehen ist.

Videoln YUV IO

Rücksetzen des Datenzeigers (Schreiben) und Übergabe der Videodaten (Lesen)

Schreibzugriff

Rücksetzen des Datenzeigers

Index: **0x0052**
Daten: **variabel**
Zulässiger Bereich: 0x0000 – 0xFFFF (keine Datenauswertung)

Lesezugriff

Übergabe der Videodaten

Index: **0x0052**
Daten: **variabel**

Anmerkungen:

Einige Betriebssysteme erlauben nur mit hohem Aufwand die Gewinnung eines Zeiges auf physikalischen Speicher. Daher besteht mit dieser Funktion die Möglichkeit, die Daten wortweise über einen I/O-Zugriff abzuholen. Vor jedem Auslesen eines Bildes muss das Rücksetzen des Lesezeigers erfolgen (das Datenwort spielt dabei keine Rolle, allein der Schreibzugriff reicht aus). Danach kann kontinuierlich mit Lesezugriffen das Videobild ausgelesen werden. Bei maximaler Auflösung käme man demnach auf 720 x 576 Übergabevorgänge (PAL). Eine Verkleinerung des Bildes bewirkt auch ein kleineres Datenaufkommen, z.B. bei einer Video-Auflösung von 200 x 150 Pixeln eben nur 200 x 150 Lesezugriffe. Eine Vergrößerung bringt hingegen keine weiteren Informationen – mehr als 720 x 576 Pixel sind real nicht vorhanden, eine Skalierung auf XGA-Format erfolgt also quasi virtuell. Das Einfrieren des Bildes geschieht hier nicht automatisch, der Programmierer hat dafür Sorge zu tragen.

HINWEIS:

Besteht die Absicht, ein Capturing durchzuführen, muss bei der Wahl der horizontalen und vertikalen Grösse des Videobildes folgende Regel zur Anwendung kommen:

Die horizontale Grösse darf nur ein Vielfaches von 8 und die vertikale Grösse ein Vielfaches von 2 betragen.

Eine Nichtbeachtung führt zu höchst unerfreulichen Ergebnissen.

Registerfunktionen

VideoIn SAA7118

Schreiben und Lesen der Register des Video-Decoders SAA7118 (PHILIPS)

Schreibzugriff

Schreiben von Daten

Index: **0x0100**

Daten: **LO-Byte - Register (variabel)**

HI-Byte - Inhalt (variabel)

Zulässiger Bereich für beide Bytes: 0x00 – 0xFF

Definition des Registers für das Lesen

Index: **0x101**

Daten: **LO-Byte - Register (variabel)**

Zulässiger Bereich: 0x00 – 0xFF

Lesezugriff

Lesen eines Registers

Index: **0x101**

Daten: **LO-Byte - Inhalt (variabel)**

Anmerkungen:

Jeder Lesezugriff benötigt die erneute Übergabe des Registers durch einen Schreibbefehl. Weitere Details bitte im Datenblatt des Video-Decoders nachlesen.

VideoIn GEODE

Schreiben und Lesen der Video-Register des Prozessors Geode SC1200 (NATIONAL/AMD)

Schreibzugriff

Schreiben von Daten

Registeranwahl (Übernahme)

Index: **0x0102**

Daten: **LO-Byte - Register (variabel)**

HI-Byte (unteres Nibble) - Register (variabel)

Bit 15

0 = F4BAR0 1 = F4BAR2

Zulässiger Bereich fürs LO/HI-Byte: 0x0000 – 0x0FFF

LO-WORD Daten

Index: **0x0103**

Daten: **variabel**

Zulässiger Bereich: 0x0000 – 0xFFFF

HI-WORD Daten

Index: **0x0104**

Daten: **variabel**

Zulässiger Bereich: 0x0000 – 0xFFFF

Definition des Registers für das Lesen

Index: **0x0105**

Daten: **LO-Byte - Register (variabel)**

HI-Byte (unteres Nibble) - Register (variabel)

Bit 15

0 = F4BAR0 1 = F4BAR2

Zulässiger Bereich fürs LO/HI-Byte: 0x0000 – 0x0FFF

Lesezugriff

Lesen eines Registers

LO-WORD Daten

Index: **0x0103**

Daten: **variabel**

HI-WORD Daten

Index: **0x0104**

Daten: **variabel**

Anmerkungen:

Jeder Lesezugriff benötigt die erneute Übergabe des Registers durch einen Schreibbefehl. Bei dem Einschreiben von Daten erfolgt die Übernahme erst mit dem Setzen des Registers. Die übliche Reihenfolge sollte also lauten: LO-WORD Daten, HI-WORD Daten und dann das Register setzen. Der Lesevorgang findet gleichfalls im Augenblick der Registerübergabe statt, somit gelten für das Lesen wie für das Schreiben die gleichen Regeln.

Weitere Details bitte im Datenblatt des Prozessors nachlesen.

Status-/Fehlerfunktionen

Videoin Status

Lesen des Interface-Status

Index: **0x0200**

Daten: **Bit 0**

0 = Video nicht initialisiert 1 = Video initialisiert

Bit 14

0 = Kein Fehler 1 = Fehler aufgetreten

Bit 15

0 = Bereit für Übergabe von Daten 1 = Nicht bereit für Übergabe

Anmerkungen:

Es ist mehr als ratsam, bei einer anstehenden Fehlermeldung den Fehlercode zu lesen, da unter Umständen bei Nichtbeachtung bestimmte Funktionen nicht ausgeführt werden.

Videoin Errorcode

Lesen des Fehlercodes

Index: **0x201**

Daten: **variabel**

Fehlercodes:

VIDEOIN_ERROR_NOERR	0x0000
VIDEOIN_ERROR_HPOS	0x0001
VIDEOIN_ERROR_VPOS	0x0002
VIDEOIN_ERROR_HSIZE	0x0003
VIDEOIN_ERROR_VSIZE	0x0004
VIDEOIN_ERROR_SCALE	0x0005
VIDEOIN_ERROR_PTR	0x0020
VIDEOIN_ERROR_ALPHA_NUM	0x0100
VIDEOIN_ERROR_ALPHA_PRIOR	0x0101
VIDEOIN_ERROR_ALPHA_HPOS	0x0102
VIDEOIN_ERROR_ALPHA_VPOS	0x0103
VIDEOIN_ERROR_ALPHA_HSIZE	0x0104
VIDEOIN_ERROR_ALPHA_VSIZE	0x0105

Anmerkungen:

Die Namensgebung dürfte schon Erläuterung genug beinhalten. Ein Fehler entsteht in der Regel bei Unter-/Überschreitung eines Grenzwertes. Die Erzeugung des Fehlercodes findet allerdings nur bei ganz bestimmten Funktionen statt, d.h. eine Parameterübergabe verursacht keine Fehlermeldung, sondern erst die Ausführung eines Kommandos (z.B. Initialisierung oder Aktivierung eines Alpha-Blending Fensters), weil erst in diesem Augenblick alle Parameter zur Verfügung stehen und eine Auswertung Sinn macht.

Anhang A

Beispiele:

Beispiel 1

Minimal-Initialisierung

```

outpw (0x0A40, 0x0001);
outpw (0x0A42, 0x0014);           // Horiz. Pos. = 20
outpw (0x0A40, 0x0002);
outpw (0x0A42, 0x000A);         // Vert. Pos. = 10
outpw (0x0A40, 0x0003);
outpw (0x0A42, 0x01F4);         // Horiz. Größe = 500
outpw (0x0A40, 0x0004);
outpw (0x0A42, 0x0190);         // Vert. Größe = 400
outpw (0x0A40, 0x0000);
outpw (0x0A42, 0x0011);         // Initialisierung
                                   // PAL, Composite
                                   // Eingangskanal 2
                                   // BOB-Deinterlacing

```

Beispiel 2

Video-Bild einfrieren mit Hilfe der Prozessor-Register

```

VOID SetFreeze (BOOL bFreeze)
{
    USHORT nDataLO, nDataHI;

    outpw (0x0A40, 0x0105);       // Lesen des Registers
    outpw (0x0A42, 0x8004);       // VIP_CONTROL in F4BAR2
    outpw (0x0A40, 0x0103);
    nDataLO = inpw (0x0A42);
    outpw (0x0A40, 0x0104);
    nDataHI = inpw (0x0A42);

    nDataLO &= 0xFFFC;
    if (bFreeze)
        nDataLO |= 0x0001;       // Bild einfrieren
    else
        nDataLO |= 0x0003;       // Bild freigeben

    outpw (0x0A40, 0x0103);
    outpw (0x0A42, nDataLO);
    outpw (0x0A40, 0x0104);
    outpw (0x0A42, nDataHI);
    outpw (0x0A40, 0x0102);
    outpw (0x0A42, 0x8004);       // Schreiben des Registers
}

```

Anhang B

Aufbau der Videodaten (Capture-Funktionen):

Grundsätzlich liegen die Daten im YUV-Format (4:2:2) vor, d.h. ein Doppelwort (32 Bit) enthält die Informationen von 2 Pixeln. Ein Doppelwort stellt sich also in folgender Weise dar (jeweils ein Byte): **Y** (1. Pixel) – **U** – **Y** (2. Pixel) – **V**.

Daraus ergibt sich als maximale Grösse des Videospeichers 720 x 576 Worte (= 829440 Bytes) für PAL und 720 x 480 Worte (= 691200 Bytes) für NTSC. Die Startadresse bleibt konstant, die Position des Videofensters übt keinen Einfluss aus. Bei Auflösungen unterhalb der Normwerte verringert sich auch die Grösse des Speicherbereiches, z.B. 400 x 300 Pixel ergeben 400 x 300 Worte (= 240000 Bytes). Da möglicherweise die beiden oben aufgeführten Verfahren zum Kopieren der Daten Schwierigkeiten bei der Umsetzung bereiten oder sich als zu langsam erweisen, nachfolgend die physikalische Startadresse des Videospeichers:

Startadresse Videospeicher = 0x409C0000

Die Umrechnung in 24-Bit RGB Daten erlauben nachstehende Formeln:

$$R = 1.1640625 (Y - 16) + 1.59375 (V - 128)$$

$$G = 1.1640625 (Y - 16) - 0.8125 (V - 128) - 0.390625 (U - 128)$$

$$B = 1.1640625 (Y - 16) + 2.015625 (U - 128)$$

Die Farbe Schwarz (RGB 0,0,0) lautet im YUV-Format 0x10801080. Die Praxis zeigt, dass eine Bereichsabsicherung stattfinden sollte, in der Art: alle Werte kleiner Null auf Null und alle Werte grösser 255 auf 255 zu setzen.

DOCUMENT REVISION HISTORY

File name	Date	Edited by	Alteration to previous document revision
ePanel_MM_V100	08/16/02	M. Hüttmann	First release
ePanel_MM_V101	02/13/03	M. Hüttmann	Basic revision
ePanel_MM_V102	04/04/03	M. Hüttmann	Add capture function
ePanel_MM_V103	07/29/04	M. Hüttmann	Add english version