AN-AL251A-EVB-A0-02

Software Application Notes for

AL251A-EVB-A0

Preliminary Rev.0 .1

Document Number: 1-E-PAE0351-0002

Amendments(Since August 14, 2006)

06.08.14 Preliminary Version 0.1

THE INFORMATION CONTAINED HEREIN IS SUBJECT TO CHANGE WIHOUT NOTICE.

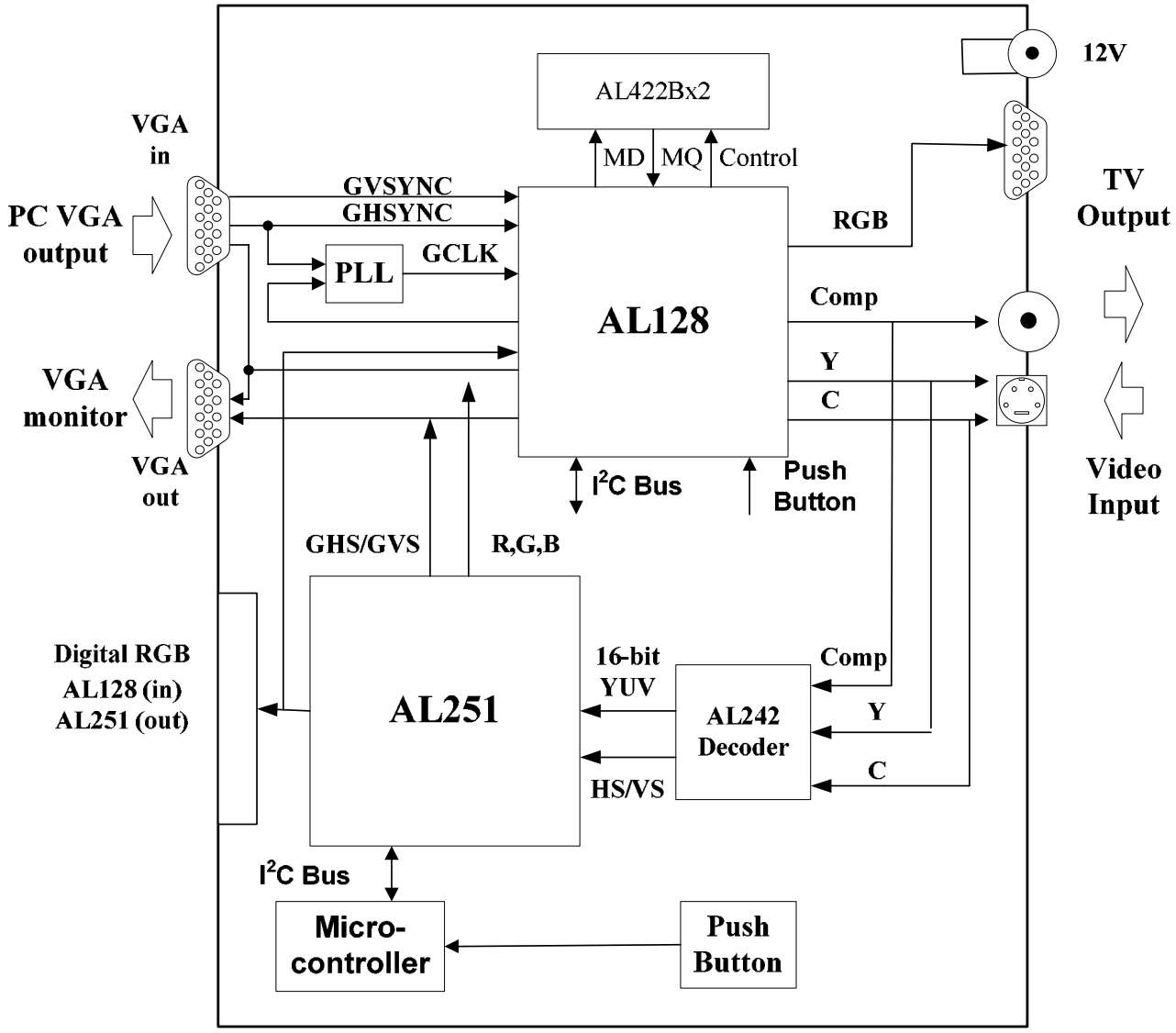
Disclaimer

THE CONTENTS OF THIS DOCUMENT ARE SUBJECT TO CHANGE WITHOUT NOTICE. AVERLOGIC TECHNOLOGIES RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HERIN TO IMPROVE RELIABILITY, FUNCTION OR DESIGN. AVERLOGIC DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HERIN; NEITHER DOSE IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS. CUSTOMERS ARE ADVISED TO CONSULT WITH AVERLOGIC OR ITS COMMERCIAL DISTRIBUTORS BEFORE ORDERING.

AN-AL251A-EVB-A1-02 Software Application Notes

Contents:

1. AL251A-EVB Block Diagram



2. General Chips Information

2.1. AL251

The AL251 Video Scan Doubler (De-Interlacer) is a video conversion chip for consumer video and multimedia applications. It converts interlaced NTSC or PAL, ITU-RBT 601 (CCIR 601) or square pixel, YUV422 or RGB565 digital signals into computer monitor RGB signals for direct connection to a computer monitor or progressive scan TV.

By using I^2C interface control, the AL251 can also be programmed to co-ordinate with various input resolutions, adjust screen positioning and crop video noise from around the original input video boundary.

The internal RGB video lookup tables (LUT), which are controlled via I2C interface, can provide gamma correction for calibrating the color accuracy of different types of CRT's and

improving the contrast level to display more vivid pictures.

A built-in on-screen-display (OSD) provides programmable bitmap RAM for custom design icons and on-screen control panels.

Overlay function is supported to create titling or on-screen-display menus for video adjustment.

The AL251 provides all the features of the AL250. Additionally, it has digital output in YUV422 or RGB565 format, and can convert NTSC video for VGA LCD panels.

The AverLogic proprietary digital signal processing technology creates a highly stable video image without tearing effects or jagged edges. The output picture is smoother and has less flicker than the original input signal/picture.

2.2. AL128

The AL128 PC to TV scan converter chip accepts graphic data up to 1024x768 resolution from PC and Macintosh graphics controllers and converts it into broadcast quality NTSC or PAL TV signals. In addition to analog RGB, 24-bit digital RGB data can be input to maintain the best video quality and avoid noise problems. This new chip is pin-to pin compatible with the AverLogic AL100 but provides analog RGB output for SCART implementation.

An integrated high-quality anti-flicker filter (SmartFilter™) removes the unpleasant flicker caused by the interlaced display of high contrast graphics while maintaining the original clarity and sharpness of informative data such as natural pictures and text. With 512Kbytes of memory, plug-and-play is achieved by automatically detecting the scan rate and resolution of the incoming graphic signals without the use of software.

With less memory than other solutions on the market, high resolution data is processed and stored by using a complex and proprietary buffer management system. No compromise is made at all with video quality by using either compression or sub-sampling algorithms.

The major functions of the AL128 can be accessed using four push buttons combined with the on-screen-menu feature, eliminating the cost of a micro-controller and complex control panel. The superior quality scaling algorithm, which reduces the jagged-edge artifacts from line dropping, can smoothly fit graphics of 640x480 (up to 100 Hz) and 1024x768 (up to 75Hz) resolutions into the visible region of the NTSC or PAL screen. Both horizontal and vertical sizes can be linearly adjusted. Additional features include eight levels of flicker control using 5-line filter, zoom control and picture freeze.

This highly integrated mix-signal chip, packaged in 24mm x 24mm 160-pin LQFP (low quad flat package), is powered by a single 5-volt power supply. Power-down is achieved by using either hardware or software control.

The enhanced features and superior quality make the AL128 very suitable for PC video to TV conversion in PC ready multimedia TV's, scan converter boxes, VGA add-on cards, Web TVs, or network / laptop PCs.

3. Programming of AL251/AL128 chips

3.1. Programming of AL251

3.1.1. Video Input

The digital video data formats that the AL251 accepts are YUV422 and RGB565. To select YUV422 or RGB565 as the input format, program the Board Configuration Register #02h, or set the hardware pin "INTYPE" (AL251 pin#14).

3.1.2. Output Formats

The AL251 provides digital output in RGB565 or YUV422 format. To select YUV422 or RGB565 as the output format, program the Control Register #08h<7>.

3.1.3. Video Timing

The AL251 registers 20h~29h and 2Bh~2Eh are used to control the video timing. All increments are either by 8 pixels per line or by 4 lines per frame. All values (times 8 or 4) are relative to the input video source H-sync or V-sync. These registers need to be programmed if the input video resolution is different from the default resolution supported.

The H-sync Start and End (registers 22h and 23h) define the output horizontal sync period relative to the input H-sync leading edge.

The Horizontal Blank Start and End (registers 2Bh and 2Ch) define the output H-sync blanking period. The Horizontal Capture Start and End (registers 20h and 21h) define the active pixels in each line relative to the input video H-sync. These registers can also be used for adjusting the position of the output picture.

The Horizontal Total High and Low (registers 24h and 29h) define the total number of pixels per line.

The AL251 can detect the H-total automatically when the input data is of the typical resolution mentioned in the Default Resolution section.

The V-sync Start and End (registers 27h and 28h) define the output V-sync period relative to the input V-sync start.

The Vertical Blank Start and End (registers 2Dh and 2Eh) define the output V-sync blanking period.

The Vertical Capture Start and End (registers 25h and 26h) define the active lines.

The total number of lines per frame (Vertical Total) is detected by the AL251 automatically. To take advantage of the auto detection of the AL251, set the bit 3 of the Control register #08h (Softtime) as 0. If a user-defined input format is used, then disable the hardware default by setting this bit as 1, and write all of the parameters to the corresponding registers to define the format.

The following typical parameters (as well as hardware default values) are for reference:

Mode	Square NTSC	Square PAL	CCIR NTSC	CCIR PAL
H(Horizontal) total	780	944	858	864
V(Vertical) total	525	625	525	625
HDE Start	120	160	72	80
HDE End	736	896	752	760
H-sync Start	776	0	792	808
H-sync End	56	72	856	16
VDE Start	48	60	48	60
VDE End	500	604	500	604
V-sync Start	4	4	4	4
V-sync End	8	8	8	8
Reg.#20h HDE Start	0Fh	14h	09h	0Ah
Reg.#21h HDE End	5Ch	70h	5Eh	5Fh
Reg.#22h H-sync Start	61h	00h	63h	65h

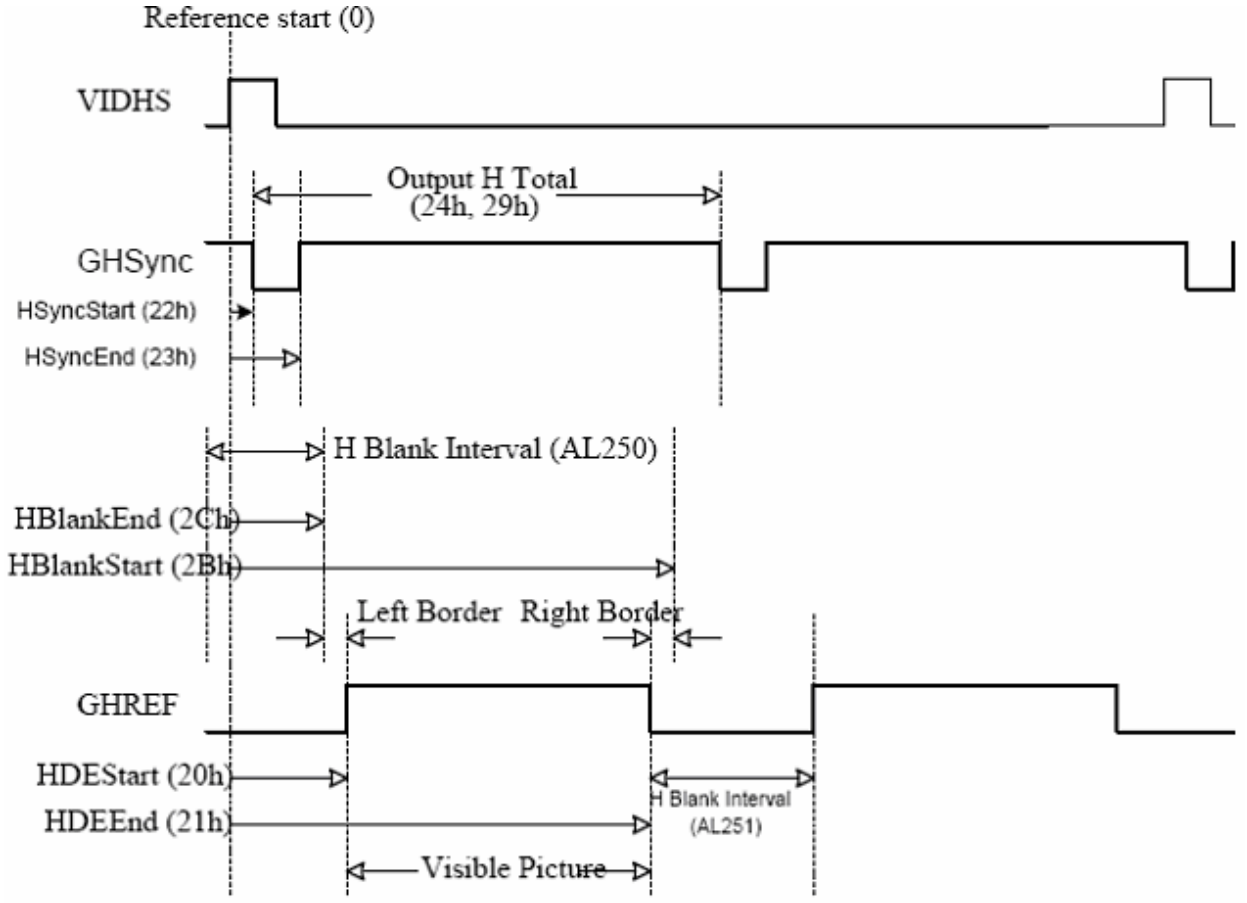
Reg.#23h H-sync End	07h	09h	6Bh	02h
Reg.#24h HTOTAL10_3	61h	76h	6Bh	6Ch
Reg.#29h HTOTAL2_1	02h	00h	01h	00h
Reg.#25h VDE Start	0Ch	0Fh	0Ch	0Fh
Reg.#26h VDE End	7Dh	97h	7Dh	97h
Reg.#27h V-sync Start	01h	01h	01h	01h
Reg.#28h V-sync End	02h	02h	02h	02h
Reg.#2Bh H-blank Start	5Fh	73h	61h	62h
Reg.#2Ch H-blank End	0Ch	11h	06h	07h
Reg.#2Dh V-blank Start	7Fh	99h	7Fh	99h
Reg.#2Eh V-blank End	0Ah	0Dh	0Ah	0Dh

The output timing/format is as follows:

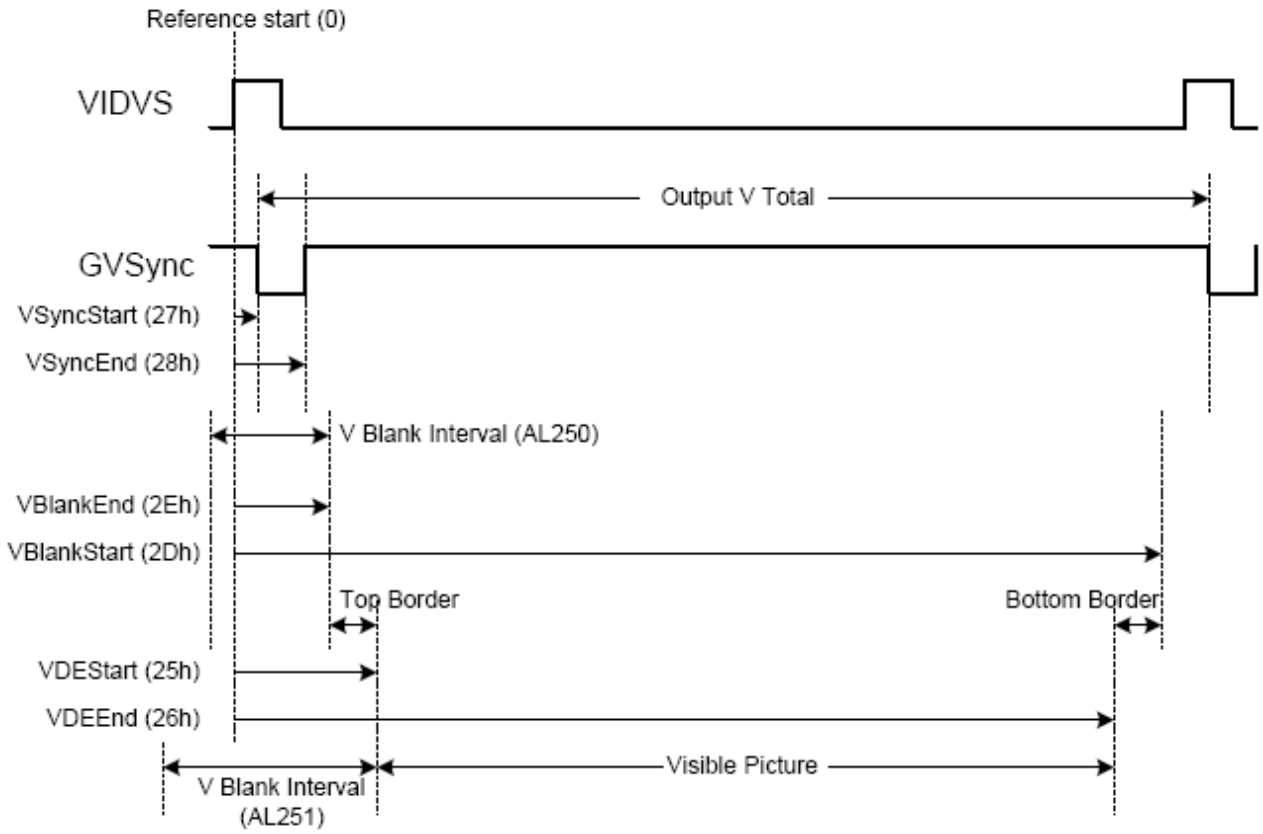
	Square NTSC	Square PAL	CCIR NTSC	CCIR PAL
Resolution	640x480/616x452	768x576/736x544	720x480/680x452	720x576/680x544
Pixel rate	24.5454 MHz	29.5 MHz	27.00 MHz	27.00 MHz
Interlace	No	No	No	No
Video	Analog-color	Analog-color	Analog-color	Analog-color
Sync on G	No	No	No	No
Video level	700mV/1V*	700mV/1V*	700mV/1V*	700mV/1V*
White level	700mV/1V*	700mV/1V*	700mV/1V*	700mV/1V*
Black level	0 IRE	0 IRE	0 IRE	0 IRE
H total	780	944	858	864
H display	616*	736*	680*	680*
H F-porch	40*	48*	40*	48*
H B-porch	64*	88*	74*	64*
HS width	60*	72*	64*	72*
H border	24*	24*	24*	24*
V total	525	625	525	625
V display	452*	544*	452*	544*
V F-porch	29*	25*	29*	25*
V B-porch	40*	52*	40*	52*
VS width	4*	4*	4*	4*
V border	8*	8*	8*	8*
HS output	ON(-)*	ON(-)*	ON(-)*	ON(-)*
VS output	ON(-)*	ON(-)*	ON(-)*	ON(-)*

Fh	31.4685 KHz	31.250 KHz	31.4685 KHz	31.250 KHz
Fv	59.94 Hz	50 Hz	59.94 Hz	50 Hz

Remark: Values with "*" are programmable (S/W) or adjustable (H/W).
The horizontal video timing diagram is as follows.



The vertical video timing diagram is as follows.



Details about the registers can be found in the Register Definition section of the datasheets.

3.1.4. Border/Border Color

The AL251 displays all the active pixels from the video source resulting in a larger viewable area on a monitor than on a regular TV. This is especially advantageous for digital video sources such as DVD. However, for some other video sources such as VCR, the unwanted and untrimmed border may appear. To solve this, the AL251 provides border control by cropping the video source. In addition, the cropped border can be filled with one color (24-bit), which is defined by registers 0Ch~0Eh.

Border/border color control applies to the AL251 analog output but not to the AL251 digital YUV/RGB output.

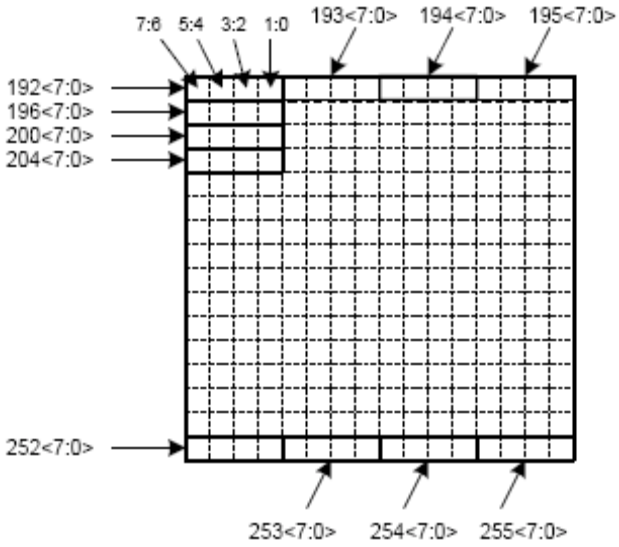
3.1.5. OSD Interface

The AL251 provides two ways to implement the on screen display. The internal way is to program the built-in on-screen display (OSD) bitmap, and the external way is to control the two overlay pins for showing on screen display or creating special effects onto each single pixel on screen.

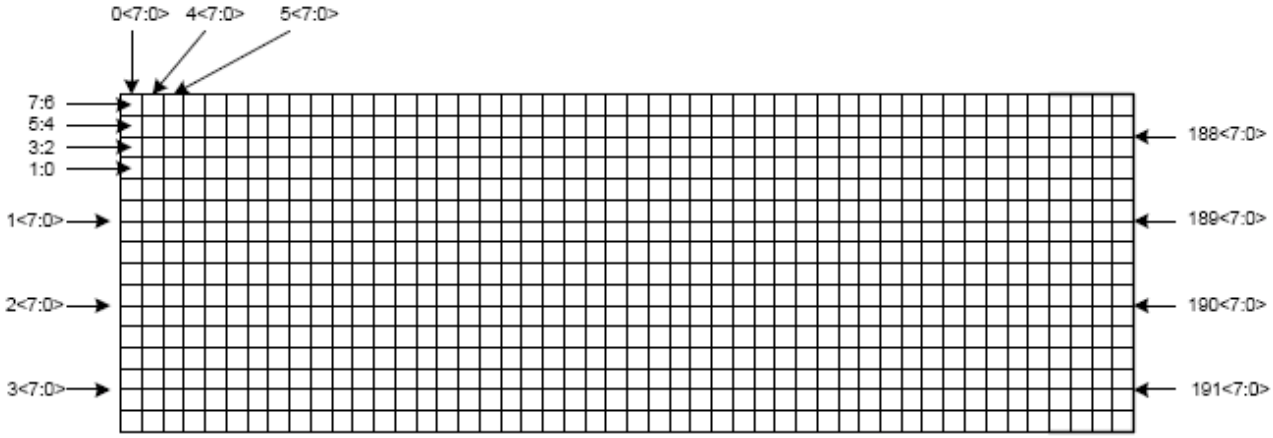
The AL251 provides 256 registers to implement the two internal bitmaps, which are programmable as 16x16 blocks (4x4 pixels each) and 48x16 blocks (8x8 pixels each) respectively.

To program the OSD, first use LUT/OSD Control register 10h to turn on bitmap 1 or bitmap 2. Then program the overlay colors 1, 2 and 3 through registers 15h~1Dh. Select the OSD index (0~255) through register 11h, then fill the data through register 13h. The two bits of each OSD block can be used to define no overlay color (transparent) or color 1, 2 or 3. Mesh color and mesh background can be enabled by programming register 2Fh. The position of the bitmaps can be defined by registers 1Eh, 1Fh, and 2Fh.

The data index of the bitmap 1 starts at bitmap address 192, and the lay-out is defined as follows:



Each pixel is defined by 2 bits value (“00”, “01”, “10” and “11”).
Value “00” shows the current input video data.
Value “01”, “10” and “11” are index to overlay color 1~3 (defined in registers 15h ~ 1Dh).
The data index of bitmap 2 starts at bitmap address 0, and the lay-out is defined as follows:



Similar to bitmap 1, each pixel is defined by 2 bits value (“00”, “01”, “10” and “11”) with the same definition.

The horizontal positions of the bitmaps 1 & 2 are defined by registers 1Eh and 1Fh respectively. The vertical position of both is defined by register 2Fh.

For the external OSD, the overlay feature needs to be used and this will be explained in detail in the External Overlay section.

OSD control applies to the AL251 analog output and the AL251 digital RGB output, but not to the AL251 digital YUV output.

3.2. Programming of AL128

3.2.1. Hardware and Software Control Modes

The AL128 is powered up to a default state depending on the hardware mode-setting pins. Eight of these hardware configuration pins are disabled by setting SoftConfig (bit 4 of register 03h) to one, and configurations are decided by the values of register 0x02, programmable by software.

By default, AL128 major functions can be controlled by On Screen Display Menu (OSDM) push buttons. The registers related to these functions have no effect on those On Screen Display Menu functions unless DisButton (bit 3 of register 03h) is set to one. When this bit is set to one, the external OSDM push buttons are deactivated and those functions are controlled by software programming. The following table lists the functions that can be controlled by either push buttons or software programming, and their corresponding registers.

Function	Register Name	Register Index
Sharpness	VFltMode	33h<2:0>
Zoom	Zoom	20h<1:0>
	ZoomEn	20h<7>
Pan	GinHStartDlt	21h<7:0>
	GinVStartDlt	22h<7:0>
	GinHStZoomDlt	2Bh<6:0>
	GinVStZoomDlt	2Ch<6:0>
Underscan	SoftUdScan	33h<6>
Position	TvHStartDlt	52h<7:0>
	TvVStartDlt	51h<7:0>
Brightness	TvBright	50h<4:3>
Color Bar	ColorBar	50h<1>

3.2.2. Pan and Position Control

The panning feature selects the portion of the graphic picture to be displayed on the TV screen. When in zoom mode, the visible screen can be progressively panned to any area of the virtual screen.

To pan the display, change the values of H-pan delta (Reg. #21h, #26h) and V-pan delta (Reg. #22h, 2Ch). To pan in zoom mode, #2Bh and #2Ch may also need to be programmed. Position control is used to center the video on the TV screen. To position the display, change the values of H-display delta (Reg. #52h) and V-display delta (Reg. #51h).

3.2.3. Zoom Feature

AL128 performs 2x zoom in both horizontal and vertical directions. This can be used with panning, or activated by jumping to any given quadrant via use of the push buttons or software. To zoom, use register #20h to enable it, and then use #2Bh and #2Ch to position it.

3.2.4. Push Button Interface/OSD

The push button interface is used with the on-screen-display menu. There are four buttons: menu, select, increment and decrement. Functions controlled by these push buttons are pan, zoom, position centering, brightness, underscan, color bar and sharpness.

The AL128 provides a user-friendly on-screen control with four push buttons, [Menu], [Select], [<] and [>], for end users to control the major functions of the AL128. There are eight onscreen control functions, which are:

Sharpness Panning Positioning Color bar
Zoom Over/underscan Brightness Home settings

- To pop up the on screen menu, press the [Menu] button.
- To select a control function, use the [<] or [>] button to scroll to it, then press the [Select] button.
- After selecting a control function, use the [Select], [<] or [>] button to adjust the control effects.
- To exit the on screen menu, press [Menu] again.

3.2.5. Programming notes for NTSC

Always set to two-field memory capture to capture odd and even field into field memory by setting reg.#02h<3:2> = "10"
Limitation: (Horizontal Capture Width) * (Vertical Capure Height) <= 384K.
*At all modes, always set Reg.#33<6> = 0 to disable the hardware default

For 640x400 (DOS)

Modes	**Underscan**	**Overscan**	**Zoom**
	Reg.#02h<3:2> = "10"	**Reg.#02h<3:2> = "10"**	**Reg.#02h<3:2> = "10"** **Reg.#20h<7> = 1**
Parameters	CapH: 720 CapV: 416->416 CapDltHST: 24 CapDltVST: 4 DspH: 720->720 DspV: 416 DspDltHST: 0 DspDltVST: 8 Hpll: 832	Same as NTSC underscan defined on the left	CapH: 736 CapV: 416->416 CapDltHST: 24 CapDltVST: 0 DspH: 368->736 DspV: 208->416 DspDltHST: 0 DspDltVST: 8 Hpll: 896

For 640x480 (VGA)

Modes	**Underscan**	**Overscan**	**Zoom**
	Reg.#02h<3:2> = "10"	**Reg.#02h<3:2> = "10"**	**Reg.#02h<3:2> = "10"** **Reg.#20h<7> = 1**
	CapH: 720 CapV: 496->432	CapH: 752 CapV: 480->480	CapH: 752 CapV: 480->480
Parameters	CapDltHST: 16 CapDltVST: -4 DspH: 720->720 DspV: 432 DspDltHST: 24 DspDltVST: 24 Hpll: 832	CapDltHST: 24 CapDltVST: 0 DspH: 752->752 DspV: 480 DspDltHST: 16 DspDltVST: 8 Hpll: 944	CapDltHST: 40 CapDltVST: 0 DspH: 376->752 DspV: 480 DspDltHST: 16 DspDltVST: 8 Hpll: 944

For 800x600 (SVGA)

Modes	**Underscan**	**Overscan**	**Zoom**
	Reg.#02h<3:2> = "10"	**Reg.#02h<3:2> = "10"**	**Reg.#02h<3:2> = "10"** **Reg.#20h<7> = 1**
Parameters	CapH: 720 CapV: 608->432 CapDltHST: 32 CapDltVST: -16 DspH: 720->720 DspV: 432 DspDltHST: 32 DspDltVST: 32 Hpll: 880	CapH: 800 CapV: 480->480 CapDltHST: 96 CapDltVST: -16 DspH: 800->800 DspV: 480 DspDltHST: 16 DspDltVST: 8 Hpll: 1056	CapH: 800 CapV: 480->480 CapDltHST: 80 CapDltVST: 0 DspH: 400->800 DspV: 480 DspDltHST: 16 DspDltVST: 8 Hpll: 1056

For 1024x768 (XGA)

Modes	**Underscan**	**Overscan**	**Zoom**
	@ < 80 Hz **Reg.#02h<3:2> = "10"**	**@ < 66 Hz** **Reg.#02h<3:2> = "10"**	**@ < 66 Hz** **Reg.#02h<3:2> = "10"** **Reg.#20h<7> = 1**
Parameters	CapH: 720 CapV: 784->432 CapDltHST: 64 CapDltVST: 0 DspH: 720->720 DspV: 432 DspDltHST: 32 DspDltVST: 32 Hpll: 864	CapH: 720 CapV: 480->480 CapDltHST: 96 CapDltVST: -4 DspH: 720->886 DspV: 480 DspDltHST: 16 DspDltVST: 8 Hpll: 1184	CapH: 720 CapV: 480->480 CapDltHST: 120 CapDltVST: 0 DspH: 360->744 DspV: 240->480 DspDltHST: 16 DspDltVST: 8 Hpll: 1184
Modes		**@ >= 66 Hz and < 73 Hz** **Reg.#02h<3:2> = "10"**	**@ >= 66 Hz and < 73 Hz** **Reg.#02h<3:2> = "10"** **Reg.#20h<7> = 1**
Parameters	Same as above	CapH: 704 CapV: 480->480 CapDltHST: 72 CapDltVST: -4 DspH: 704->866 DspV: 480 DspDltHST: 16 DspDltVST: 8 Hpll: 1024	CapH: 720 CapV: 480->480 CapDltHST: 80 CapDltVST: 0 DspH: 360->794 DspV: 240->480 DspDltHST: 16 DspDltVST: 8 Hpll: 1024
Modes		**@ >= 73 Hz and < 80 Hz** Reg.#02h<3:2> = "10"	**@ >= 73 Hz and < 80 Hz** Reg.#02h<3:2> = "10" Reg.#20h<7> = 1

Parameters	Same as above	CapH: 704 CapV: 480->480 CapDltHST: 32 CapDltVST: -4 DspH: 704->900 DspV: 480 DspDltHST: 16 DspDltVST: 8 Hpll: 928	CapH: 656 CapV: 480->480 CapDltHST: 80 CapDltVST: 0 DspH: 328->778 DspV: 240->480 DspDltHST: 16 DspDltVST: 8 Hpll: 928
Modes	**@ >= 80 Hz** **Reg.#02h<3:2> = "10"**	**@ >= 80 Hz** **Reg.#02h<3:2> = "10"**	**@ >= 80 Hz** **Reg.#02h<3:2> = "10"** **Reg.#20h<7> = 1**
Parameters	CapH: 720 CapV: 784->432 CapDltHST: 56 CapDltVST: 0 DspH: 720->744 DspV: 432 DspDltHST: 32 DspDltVST: 32 Hpll: 832	CapH: 640 CapV: 480->480 CapDltHST: 32 CapDltVST: -4 DspH: 640->890 DspV: 480 DspDltHST: 16 DspDltVST: 8 Hpll: 832	CapH: 640 CapV: 480->480 CapDltHST: 80 CapDltVST: 0 DspH: 320->836 DspV: 240->480 DspDltHST: 16 DspDltVST: 8 Hpll: 832

Index:

CapH: Horizontal Capture Width
CapV: Vertical Capture Height
CapDltHST: Horizontal Capture Delta Start
CapDltVST: Vertical Capture Delta Start
DspH: Horizontal Display Width
DspV: Vertical Display Height
DspHST: Horizontal Display Start
DspVST: Vertical Display Start
Hpll: Horizontal Total
->: scaled to

3.2.6. Programming notes for PAL

Always set to two-field memory capture to capture odd and even field into field memory by setting reg.#02h<3:2> = "10"
Limitation: (Horizontal Capture Width) * (Vertical Capure Height) <= 384K.
*At all modes, always set Reg.#33<6> = 0 to disable the hardware default

For 640x400 (DOS)

	Underscan	**Overscan**	**Zoom**
Modes	**Reg.#02h<3:2> = "01"** **Reg.#40 = 00h** **Reg.#53<4> = 0**	**Reg.#02h<3:2> = "01"** **Reg.#40 = 00h** **Reg.#53<4> = 0**	**Reg.#02h<3:2>="01"** **Reg.#20h<7> = 1** **Reg.#40 = 00h** **Reg.#53<4> = 0**
Parameters	CapH: 800 CapV: 416->416 CapDltHST: 24 CapDltVST: 4 DspH: 800->800 DspV: 416 DspDltHST: -48 DspDltVST: 0 Hpll: 960	Same as PAL underscan defined on the left	CapH: 800 CapV: 416->416 CapDltHST: 40 CapDltVST: 8 DspH: 400->852 DspV: 208->416 DspDltHST: -48 DspDltVST: 0 Hpll: 896

For 640x480 (VGA)

Modes	Underscan	Overscan	Zoom
	Reg.#02h<3:2> = ”01” **Reg.#40 = 00h** **Reg.#53<4> = 0**	**Reg.#02h<3:2> = ”01”** **Reg.#40 = 00h** **Reg.#53<4> = 0**	**Reg.#02h<3:2> = ”01”** **Reg.#20h<7> = 1** **Reg.#40 = 00h** **Reg.#53<4> = 0**
Parameters	CapH: 800 CapV: 496->496 CapDltHST: 16 CapDltVST: -4 DspH: 800->800 DspV: 496 DspDltHST: -56 DspDltVST: -12 Hpll: 992	Same as PAL underscan defined on the left	CapH: 800 CapV: 496->496 CapDltHST: 48 CapDltVST: -4 DspH: 400->800 DspV: 248->496 DspDltHST: -56 DspDltVST: -12 Hpll: 992

For 800x600 (SVGA)

Modes	Underscan	Overscan	Zoom
	Reg.#02h<3:2> = ”01” **Reg.#40 = 00h** **Reg.#53<4> = 0**	**Reg.#02h<3:2> = ”01”** **Reg.#40 = 00h** **Reg.#53<4> = 0**	 **Reg.#20h<7> = 1** **Reg.#40 = 00h** **Reg.#53<4> = 0**
Parameters	CapH: 800 CapV: 608->528 CapDltHST: 8 CapDltVST: -12 DspH: 800->800 DspV: 528 DspDltHST: 88 DspDltVST: 32 Hpll: 1056	CapH: 800 CapV: 576->576 CapDltHST: 0 CapDltVST: -12 DspH: 800->882 DspV: 576 DspDltHST: 64 DspDltVST: 28 Hpll: 1056	CapH: 800 CapV: 576->576 CapDltHST: 16 CapDltVST: 0 DspH: 400->832 DspV: 288->576 DspDltHST: 64 DspDltVST: 28 Hpll: 1056

For 1024x768 (XGA)

Modes	**Underscan**	**Overscan**	**Zoom**
	@ < 66 Hz **Reg.#02h<3:2> = ”01”** **Reg.#40 = 00h** **Reg.#53<4> = 0**	**@ < 66Hz** **Reg.#02h<3:2> = ”10”** **Reg.#40 = 81h** **Reg.#53<4> = 1**	**@ < 66 Hz** **Reg.#20h<7> = 1** **Reg.#02h<3:2> = ”01”** **Reg.#40 = 00h** **Reg.#53<4> = 0**
Parameters	CapH: 800 CapV: 784->528 CapDltHST: 24 CapDltVST: -4 DspH: 800->800 DspV: 528 DspDltHST: 88 DspDltVST: 40 Hpll: 1024	CapH: 720 CapV: 544->544 CapDltHST: 96 CapDltVST: 0 DspH: 720->886 DspV: 544 DspDltHST: 64 DspDltVST: 28 Hpll: 1184	CapH: 800 CapV: 576->576 CapDltHST: 48 CapDltVST: 0 DspH: 400->800 DspV: 288->576 DspDltHST: 64 DspDltVST: 28 Hpll: 1184
	@ >= 66 Hz and < 73 Hz	**@ >= 66Hz and < 73 Hz**	**@ >= 66 Hz and < 73 Hz**
Modes	**Reg.#02h<3:2> = ”01”** **Reg.#40 = 00h Reg.#53<4> = 0**	**Reg.#02h<3:2> = ”10”** **Reg.#40 = 81h Reg.#53<4> = 1**	**Reg.#20h<7> = 1** **Reg.#02h<3:2> = ”01”** **Reg.#40 = 00h Reg.#53<4> = 0**
Parameters	CapH: 800 CapV: 784->528 CapDltHST: 0 CapDltVST: -4 DspH: 800->825 DspV: 528 DspDltHST: 88 DspDltVST: 40 Hpll: 1008	CapH: 720 CapV: 544->544 CapDltHST: 40 CapDltVST: 0 DspH: 720->903 DspV: 544 DspDltHST: 64 DspDltVST: 28 Hpll: 1024	CapH: 800 CapV: 576->576 CapDltHST: 8 CapDltVST: 0 DspH: 400->825 DspV: 288->576 DspDltHST: 64 DspDltVST: 28 Hpll: 1024
	@ >= 73 Hz and < 80 Hz	**@ >= 73Hz and < 80 Hz**	**@ >= 73 Hz and < 80 Hz**
Modes	**Reg.#02h<3:2> = ”01”** **Reg.#40 = 00h Reg.#53<4> = 1**	**Reg.#02h<3:2>=”10”** **Reg.#40 = 81h Reg.#53<4> = 1**	**Reg.#20h<7> = 1** **Reg.#02h<3:2> = ”01”** **Reg.#40 = 00h Reg.#53<4> = 0**
Parameters	CapH: 800 CapV: 784->528 CapDltHST: 0 CapDltVST: -4 DspH: 800->853 DspV: 528 DspDltHST: 88 DspDltVST: 40 Hpll: 928	CapH: 720 CapV: 544->544 CapDltHST: 32 CapDltVST: 0 DspH: 720->960 DspV: 544 DspDltHST: 64 DspDltVST: 28 Hpll: 928	CapH: 752 CapV: 576->576 CapDltHST: 16 CapDltVST: 0 DspH: 376->1046 DspV: 288->576 DspDltHST: 64 DspDltVST: 28 Hpll: 928
	@ >= 80 Hz	**@ >= 80Hz**	**@ >= 80 Hz**
Modes	**Reg.#02h<3:2> = ”01”** **Reg.#40 = 00h Reg.#53<4> = 1**	**Reg.#02h<3:2> = ”10”** **Reg.#40 = 81h Reg.#53<4> = 1**	**Reg.#20h<7> = 1** **Reg.#02h<3:2> = ”01”** **Reg.#40 = 00h Reg.#53<4> = 0**
Parameters	CapH: 800 CapV: 784->528 CapDltHST: -48 CapDltVST: -4 DspH: 800->1024 DspV: 528 DspDltHST: 88 DspDltVST: 40 Hpll: 832	CapH: 720 CapV: 544->544 CapDltHST: 8 CapDltVST: 0 DspH: 720->1047 DspV: 544 DspDltHST: 64 DspDltVST: 28 Hpll: 832	CapH: 752 CapV: 576->576 CapDltHST: -8 CapDltVST: 0 DspH: 376->1203 DspV: 288->576 DspDltHST: 64 DspDltVST: 28 Hpll: 832

Index: Same as NTSC

4. AL251/AL242 EVB Sample Code

4.1. Compiler

Keil C u 2.32 or newer, C51 7.02b or newer.

4.2. Source Code Descriptions

4.2.1. main.c

Main program.

void main (void);
Description: This function calls
a. AL242Initialize(); to initialize AL242.
b. AL251Initialize(); to initialize AL251
c. AutoDetectProgress(); to detect video standard of input video signals.
d. dblAutoDetect(); to detect .al251's input mode.
etc.
Input parameters: None.
Output parameters: None.
Return values: None.

4.2.2. 242.c

Group of AL242 setup and AutoDetect API, mostly called from Main();;

void InitReg(unsigned char *pcDat);
Description: Initialize AL242 register:
Input parameters:
pcDat: pointer to the data stream will be transmit.
Output parameters: None.
Return values: None.

void SetAl240Standard(unsigned char cStd);
Description: set AL242 video standard.
Input parameters: **cStd:**

STD_NTSC_M	0
STD_NTSC_443	1
STD_PAL_I	2
STD_PAL_M	3
STD_PAL_CN	4
STD_SECAM	5
STD_PAL_60	6
STD_BW625	7
STD_BW525	8

Output parameters: None.
Return values: None.

void GetSignalStatus(void);

Description: get status of current video signal.
Input parameters: None.
Output parameters:
cSignalstatus;.
Return values: None.

char CheckModeLock(void);
Description: check detect mode if locked.
Input parameters: None.
Output parameters: None
Return values: true or false.

void AfeAutoGainControl(void);
Description: AL242 Afe AGC.
Input parameters: None.
Output parameters: None.
Return values: None.

void StandardDetect(void);
Description: video standard detect.
Input parameters: None.
Output parameters: None.
Return values: None.

void UpdateStandard(void)
Description: Update Standard.
Input parameters: None:
Output parameters: None.
Return values: None.

unsigned char InputChannelSelectAL242(unsigned char _InputChannelX)
Description: select AL242 input channe
Input parameters: _InputChannelX:
0 AV
1 SV
Output parameters: None.
Return values: None.

void AutoDetectProgress(void)
Description: video standard detect and setup.
Input parameters: None.
Output parameters: None.
Return values: None.

void AL242Initialize(void)
Description: initialize AL242.
Input parameters: None.
Output parameters: None.
Return values: None.

void dcMoreBrightness()
Description: increase brightness
Input parameters: None.
Output parameters: None.
Return values: None.

void dcLessBrightness()
Description: decrease brightness.
Input parameters: None.
Output parameters: None.
Return values: None.

void dcMoreContrast()
Description: increase contrast.
Input parameters: None.
Output parameters: None.
Return values: None.

void dcLessContrast()
Description: decrease contrast.
Input parameters: None.
Output parameters: None.
Return values: None.
void dcMoreSaturation()
Description: increase saturation.
Input parameters: None.
Output parameters: None.
Return values: None.

void dcLessSaturation()
Description: decrease saturation.
Input parameters: None.
Output parameters: None.
Return values: None.

void dcMoreHue()
Description: increase hue.
Input parameters: None.
Output parameters: None.
Return values: None.

void dcLessHue()
Description: decrease hue.
Input parameters: None.
Output parameters: None.
Return values: None.

unsigned char dcGetBrightness();
Description: get current brightness value.
Input parameters: None.

Output parameters: None.
Return values: current brightness value.

unsigned char dcGetContrast();
Description: get current contrast value.
Input parameters: None.
Output parameters: None.
Return values: current contrast value.

unsigned char dcGetSaturation();
Description: get current saturation value.
Input parameters: None.
Output parameters: None.
Return values: current saturation value.

unsigned char dcGetHue();
Description: get current hue value.
Input parameters: Nonr.
Output parameters: None.
Return values: current hue value.

4.2.3. AL251.c

unsigned char Get_AL251Register(unsigned char index);
Description: get AL251 register value from index address
Input parameters: index : register address.
Output parameters: None.
Return values: register value.

void Set_AL251Register(unsigned char index, unsigned char value);

Description: write AL251 register value to index address.
Input parameters:
index: register address.
value: data will be write.
Output parameters: None.
Return values: None.

void al251Initialize(void);
Description: initialize AL251.
Input parameters: None.
Output parameters: None.
Return values: None.

void alDelay(unsigned int time_ms);
Description: delay some ms.
Input parameters: time_ms: value of delay time.
Output parameters: None.
Return values: None.

unsigned char dblAutoDetect(void)
Description: detect AL251 input mode.
Input parameters: None.
Output parameters: None.
Return values: 0: NTSC 1: PAL

void exit(void);
Description: AL251 exit, hide OSD and disable color LUT.
Input parameters: None.
Output parameters: None.
Return values: None.

void dblSetSystem(unsigned char system, unsigned char crop);
Description: AL251 system setup, setup video capture control timing and blank timing.
Input parameters: system: video input format 0--NTSC; 1--PAL.
crop: adjust value. adjust video capture size.
Output parameters: None.
Return values: None.

void dblSetOverlayColor(unsigned char i, unsigned char r0, unsigned char g0, unsigned char b0)
Description: AL251 setup overlay color
Input parameters: i: select overlay color group 0--color1 1--color2 2--color3
r0,g0,b0: tricolor value, use to adjust overlay color.
Output parameters: None.
Return values: None.

void show_input(unsigned char source)
Description: show input OSD , video source indication
Input parameters: source: video source select 0--composite 1--S-video
Output parameters: None.
Return values: None.

void osdmenu(unsigned int osd_count, unsigned char value)
Description: menu OSD show and command
Input parameters: osdcount: OSD counter, indicate four parameters. example: brightness, contrast, saturation and hue.
value: value of selected parameter
Output parameters: None.
Return values: None.

void show_osd(unsigned int osdcount)
Description: show menu OSD
Input parameters: osdcount: OSD counter, indicate four parameters. example: brightness, contrast, saturation and hue.
Output parameters: None.
Return values: None.

void show_value(unsigned char value)
Description: show OSD parameters value

Input parameters: value: need display it on the screen what the parameters value.
Output parameters: None.
Return values: None.

void OnBorder(void)
Description: the border adjust and setup border color.
Input parameters: None.
Output parameters: None.
Return values: None.

void dblSetBorderColor(unsigned char r, unsigned char g, unsigned char b)
Description: setup border color
Input parameters: r, g, b: tricolor value, may compose into others color.
Output parameters: None.
Return values: None.

void OnBootLut(void)
Description: gamma correction adjust by the [LUT] button.
Input parameters: None.
Output parameters: None.
Return values: None.

void dblBootLut(unsigned char R, unsigned char G, unsigned char B)
Description: LUT table write
Input parameters: r, g, b: RGB value for LUT arithmetic.
Output parameters: None.
Return values: None.

void dlSinBootLut(unsigned char boot)
Description: gamma correction LUT arithmetic
Input parameters: boot: adjust value for LUT arithmetic.
Output parameters: None.
Return values: None.

4.2.4. AL242i2c.c

I2c.c
Collections of AL242 video decoder and AL251 video converter I2C API.

CONTACT INFORMATION

Averlogic Technologies, Corp.
URL: http://www.averlogic.com