

AmigaMail

COLLABORATORS

	<i>TITLE :</i> AmigaMail		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		July 19, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	AmigaMail	1
1.1	V-1: Using the ECS Genlocking Features	1

Chapter 1

AmigaMail

1.1 V-1: Using the ECS Genlocking Features

By Ewout Walraven

The ECS Denise chip (8373-R2a), coupled with the release 2.0 graphics.library, opens up a whole new set of genlocking possibilities. Unlike the old Denise, whose only genlocking ability allowed keying on color register zero, the ECS Denise allows keying on any color register. Also, the ECS Denise allows keying on any bitplane of the ViewPort being genlocked. With the ECS Denise, the border area surrounding the display can be made transparent (always passes video) or opaque (overlays using color 0). All the new features are set individually for each ViewPort. These features can be used in conjunction with each other, making interesting scenarios possible. These hardware capabilities and the software support in release 2.0 offer new challenges for video oriented software.

VideoControl()

The graphics.library function VideoControl() modifies the operation of a ViewPort's ColorMap and also reports on the ColorMap's status.

```
error = BOOL VideoControl( struct ColorMap *colormap,  
                           struct TagItem *videocommandstagarray );
```

Using VideoControl(), a program can enable, disable, or obtain the state of a ViewPort's genlocking features. It returns NULL if no error occurred. The function uses a tag based interface.

This article is concerned with the following VideoControl() tags:

```
VTAG_BITPLANEKEY_GET  
VTAG_BITPLANEKEY_SET  
VTAG_BITPLANEKEY_CLR  
VTAG_CHROMA_PLANE_GET  
VTAG_CHROMA_PLANE_SET  
VTAG_BORDERBLANK_GET
```

```
VTAG_BORDERBLANK_SET
VTAG_BORDERBLANK_CLR
VTAG_BORDERNOTRANS_GET
VTAG_BORDERNOTRANS_SET
VTAG_BORDERNOTRANS_CLR
VTAG_CHROMAKEY_GET
VTAG_CHROMAKEY_SET
VTAG_CHROMAKEY_CLR
VTAG_CHROMAPEN_GET
VTAG_CHROMAPEN_SET
VTAG_CHROMAPEN_CLR
```

See <graphics/videocontrol.h> for a complete list of all the available tags.

VTAG_BITPLANEKEY_GET is used to find out the status of the bitplane keying mode. VTAG_BITPLANEKEY_SET and VTAG_BITPLANEKEY_CLR activate and deactivate bitplane keying mode. If bit plane key mode is on, genlocking will key on the bits set in a specific bitplane from the ViewPort (the specific bitplane is set with a different tag). The data portion of these tags is NULL.

For inquiry commands like VTAG_BITPLANEKEY_GET (tags ending in _GET), VideoControl() changes the _GET tag ID to the corresponding _SET or _CLR tag ID, reflecting the current state. For example, when passed the following tag array:

```
struct TagItem videocommands[] =
{
    { VTAG_BITPLANEKEY_GET, NULL},
    {VTAG_END_CM, NULL}
};
```

VideoControl() changes the VTAG_BITPLANEKEY_GET ID to VTAG_BITPLANEKEY_SET if bit plane keying is currently on, or to VTAG_BITPLANEKEY_CLR if bit plane keying is off. In both of these cases, VideoControl() only uses the tag's ID, ignoring the tag's data field.

The VTAG_CHROMA_PLANE_GET tag returns the number of the bitplane keyed on when bit plane keying mode is on. VideoControl() changes the tag's data value to the bitplane number. VTAG_CHROMA_PLANE_SET sets the bitplane number to the tag's data value.

VTAG_BORDERBLANK_GET is used to obtain the border blank mode status. This tag works exactly like VTAG_BITPLANEKEY_GET. VideoControl() changes the tag's ID to reflect the current border blanking state. VTAG_BORDERBLANK_SET and VTAG_BORDERBLANK_CLR activate and deactivate border blanking. If border blanking is on, the Amiga will not display anything in its display border, allowing an external video signal to show through the border area. On the Amiga display, the border appears black. The data portion of these tags is NULL.

Respectively, the VTAG_BORDERNOTRANS_GET, VTAG_BORDERNOTRANS_SET, and VTAG_BORDERNOTRANS_CLR tags are used to obtain the status of BorderNoTransparent mode, and to activate and to deactivate this mode. If set, the Amiga display's border will overlay external video with

the color in register 0. Because border blanking mode takes precedence over BorderNoTransparent mode, setting BorderNoTransparent has no effect if border blanking is on. The data portion of these tags is NULL.

Respectively, the VTAG_CHROMAKEY_GET, VTAG_CHROMAKEY_SET, and VTAG_CHROMAKEY_CLR tags are used to obtain the status of chroma keying mode, and to activate and deactivate chroma keying mode. If set, the genlock will key on colors from specific color registers (the specific color registers are set using a different tag). If chroma keying is not set, the genlock will key on color register 0. The data portion of these tags is NULL.

VTAG_CHROMAPEN_GET obtains the chroma keying status of an individual color register. The tag's data field contains the register number. Like the other _GET tags, VideoControl() changes the tag ID to one reflecting the current state. VTAG_CHROMAPEN_SET and VTAG_CHROMAPEN_CLR activate and deactivate chroma keying for each individual color register. chroma keying can be active for more than one register. By turning off border blanking and activating chroma keying mode, but turning off chroma keying for each color register, a program can overlay every part of an external video source, completely blocking it out.

After using VideoControl() to set values in the ColorMap, the ColorMap's ViewPort has to be rebuilt with MakeVPort(), MrgCop(), and LoadView(), so the changes can take effect. A program that uses a screen's ViewPort rather than its own ViewPort should use the Intuition functions MakeScreen() and RethinkDisplay() to make the display changes take effect.

The following example, genlockdemo.c, shows how to add genlock support for an Intuition screen.