AlphaVision™ PC Picture Box Control Protocol

Manual part number: 9711-2706

Revision date: 6/3/2002

ADAPTIVE®

NOTE: Due to continuing product innovation, specifications in this manual are subject to change without notice.

Adaptive Micro Systems
7840 North 86th Street
Milwaukee, WI 53224 USA
414-357-2020
414-357-2029 (fax)
http://www.adaptivedisplays.com

Contents

Revision history

Revision	Date	Notes
9711-2706	6/1/2002	First release

PictureBox Control Methods

Circle

Syntax:

Drawing an outline of a circle:

AVPCPictureBox.DrawCircle(c*X* as integer, c*Y* as integer, *Radius* as integer, *Color* as AMS_Colors, *ArcStart* as single, *ArcEnd* as single, *Aspect* as single)

Parameters:

cX – Integer

The value indicates the x-coordinate for the center point of the circle, ellipse, or arc. The unit of measure used is pixels.

cY – Integer

The value indicates the y-coordinate for the center point of the circle, ellipse, or arc. The unit of measure used is pixels.

Radius – Integer

The value indicates the radius of the circle, ellipse, or arc, in pixels.

CircleColor – Optional AMS_Colors – Black, Red, Green, Amber

The value indicates the RGB color of the circle's outline. If omitted, the color will be defaulted to the current forecolor of the picture box. (The only colors available on the display are Black/Off, Red, Green, and Amber.)

ArcStart - Optional Single

When an arc or a partial circle or ellipse is drawn, *start* and *end* specify (in radians) the beginning and end positions of the arc. The range for both is –2 pi radians to 2 pi radians. The default value for *start* is 0 radians; the default for *end* is 2 * pi radians.

ArcEnd – Optional Single

When an arc or a partial circle or ellipse is drawn, *start* and *end* specify (in radians) the beginning and end positions of the arc. The range for both is –2 pi radians to 2 pi radians. The default value for *start* is 0 radians; the default for *end* is 2 * pi radians.

Aspect – Optional Single

The value indicates the aspect ratio of the circle. The default value is 1.0, which yields a perfect (non-elliptical) circle on any screen

Remarks:

Method will draw a circle, elipse or arc in the picturebox of the control.

Example:

AVPCPictureBox1.DrawCircle 11, 22, 3, AMBER

This will draw an amber color circle with the center point at coordinate (11,22) and has a radius of 3 pixels.

Clear

Syntax:

AVPCPictureBox.Clear

Remarks:

This method clears the picture box of its contents.

Example:

AVPCPictureBox1.Clear

This will clear the picture box.

DrawPoint

Syntax:

AVPCPictureBox.DrawPoint(pX as integer, pY as integer, Color as AMS_Colors)

Parameters:

pX – Integer

The value indicates the x-coordinate of the point to set.

pY – Integer

The value indicates the y-coordinates of the point to set.

PointColor – Optional AMS_Colors – Black, Red, Green, Amber

The value indicates the RGB color of the point. If omitted, the color will be defaulted the current forecolor of the picture box. (The only colors available on the display are Black/Off, Red, Green, and Amber.)

Remarks:

Method will set a single pixel to a specific color.

Example:

AVPCPictureBox1.DrawPoint 11, 22, AMBER

This will set the pixel at coordinate (11,22) to color Amber.

DrawText

Syntax:

AVPCPictureBox.DrawText(curX as integer, curY as integer, Newtext as string, Color as ASM_Colors)

Parameters:

curX – Integer

Sets Current X position for text

curY – Integer

Sets Current Y position for text

Newtext – String

This is the text to be printed in the PictureBox.

TextColor – Optional AMS_Colors – Black, Red, Green, Amber

The value indicates the RGB color of the text. If omitted, the color will be defaulted to the current forecolor of the picture box. (The only colors available on the display are Black/Off, Red, Green, and Amber.)

Remarks:

Method will write text in the picture box. The text will be written starting at the point designated by the CurrentX and CurrentY. Furthermore, the text will have the attributes as designated by the various properties listed below.

Example:

AVPCPictureBox1.DrawText 11, 22, "hello", AMBER

This will draw the text "hello" with the top left most pixel at coordinate (11,22).

Line

Syntax:

Drawing a Line

AVPCPictureBox.DrawLine(*x1* as integer, *y1* as integer, *x2* as integer, *y2* as integer, *Color* as AMS_Colors)

Drawing a Rectangle

AVPCPictureBox.DrawRectangle(*x1* as integer, *y1* as integer, *x2* as integer, *y2* as integer, *Color* as AMS_Colors)

Drawing a Solid Rectangle

AVPCPictureBox.DrawFillRectangle (*x1* as integer, *y1* as integer, *x2* as integer, *y2* as integer, *Color* as AMS_Colors)

Parameters:

x1 – Integer

The value indicates the x-coordinate of the starting point for the line or rectangle. The unit of measure used is pixels.

y1 – Integer

The value indicates the y-coordinate of the starting point for the line or rectangle. The unit of measure used is pixels.

x2 – Integer

The value indicates the x-coordinate of the end point for the line or rectangle. The unit of measure used is pixels.

y2 – Integer

The value indicates the y-coordinate of the end point for the line or rectangle. The unit of measure used is pixels.

LineColor – Optional ASM_Colors – Black, Red, Green, Amber

The value indicates the RGB color of the line. If omitted, the color will be defaulted to the current forecolor of the picture box. (The only colors available on the display are Black/Off, Red, Green, and Amber.)

Remarks:

Method will draw a line, a rectangle, and a solid rectangle in the picturebox of the control.

Examples:

AVPCPictureBox1.DrawLine 0, 0, 11, 12, GREEN

This will draw a straight green line starting from coordinate (0,0) and ending as (11,22).

AVPCPictureBox1.DrawRectangle 0, 0, 11, 22, GREEN

This will draw a hollow green rectangle with the upper left corner at coordinate (0,0) and the lower right corner at (11,22).

AVPCPictureBox1.DrawFillRectangle 0, 0, 11, 11, GREEN

This will draw a solid green rectangle with the upper left corner at coordinate (0,0) and the lower right corner at (11,22).

SendToSign

Syntax:

AVPCPictureBox.SendToSign

Remarks:

Sends the current AVPCPictureBox display to the marquee using the settings set by ConnectMode property.

Example:

AVPCPictureBox1.SendToSign

PictureBox Control Properties

BackColor

Type: AMS_Colors – Black, Red, Green, Amber

Availability: Design-time or Run-time

Remarks:

Sets the background color of the picture box.

Example:

AVPCPictureBox1.BackColor = RED

This will set the background color to red.

CurrentX

Type: Integer

Availability: Run-time only

Remarks:

Returns the horizontal coordinate for the next printing or drawing method. This is read only.

Example:

OldX = AVPCPictureBox1.CurrentX

CurrentY

Type: Integer

Availability: Run-time only

Remarks:

Returns the vertical coordinate for the next printing or drawing method. This is read only.

Example:

OldY = AVPCPictureBox1.CurrentY

DrawWidth

Type: Integer

Availability: Design-time or Run-time

Remarks:

Set the line thickness for lines, circles, and squares.

Example:

AVPCPictureBox1.DrawWidth = 3

This will set the width of all lines drawn to 3 pixels wide.

FillColor

Type: AMS_Colors – Black, Red, Green, Amber

Availability: Design-time or Run-time

Remarks:

Sets the color used to fill in shapes, for example, circles and boxes created with the **DrawCircle** and **DrawRectangle** methods.

Example:

AVPCPictureBox1.FillColor = GREEN

This will fill the circle or square with green pattern defined by **FillStyle**.

FillStyle

Type: Integer

Availability: Design-time or Run-time

Remarks:

Sets the pattern used to fill shape controls as well as circles and boxes created with the **DrawCircle** and **DrawRectangle** methods. The fill style can be solid, transparent (default), horizontal line, vertical line, upward diagonal, downward diagonal, cross, or diagonal cross.

Example:

AVPCPictureBox1.FillStyle = 3

This will fill the circle or square with vertical lines.

FontBold

Type: Boolean

Availability: Design-time or Run-time

Remarks:

Sets the font to display in bold text.

Example:

AVPCPictureBox1.FontBold = True

This will set the font to bold.

FontItalic

Type: Boolean

Availability: Design-time or Run-time

Remarks:

Sets the font to display in italic text.

Example:

AVPCPictureBox1.FontItalic = True

This will set the font to italic.

FontName

Type: String

Availability: Design-time or Run-time

Remarks:

Sets the font that will be displayed.

Example:

AVPCPictureBox1.FontName = "Arial"

This will use Arial font.

FontSize

Type: Integer

Availability: Design-time or Run-time

Remarks:

Sets the size of the font to be used for text displayed in the picture boxes run-time drawing or printing operation.

Example:

AVPCPictureBox1.FontSize = 10

This will set the font size to 10.

FontStrikethru

Type: Boolean

Availability: Design-time or Run-time

Remarks:

Sets the font to display text with strike-through.

Example:

AVPCPictureBox1.FontStrikethru = True

This will stikethru the current font.

FontTransparent

Type: Boolean

Availability: Design-time or Run-time

Remarks:

If true, the text will display text with a transparent background. If false, it will display text with a solid background. The background is the same color as set with the BackColor control.

Example:

AVPCPictureBox1.FontTransparent = True

This will display the text over the existing drawing/background.

FontUnderline

Type: Boolean

Availability: Design-time or Run-time

Remarks:

Sets the font to display underlined text.

Example:

AVPCPictureBox1.FontUnderline = True

This will underline the current font.

ForeColor

Type: AMS_Colors – Black, Red, Green, Amber

Availability: Design-time

Remarks:

Sets the foreground color used to display text and graphics in the picture box.

Example:

AVPCPictureBox1.ForeColor = AMBER

This will set the fore color to amber.

ScaleHeight

Type: Integer

Availability: Run-time Only

Remarks:

Returns the number of pixels for the vertical measurement of the interior of the picture box. This is a read only control.

Example:

Height = AVPCPictureBox1.ScaleHeight

This will set the variable "Height" equal to the height of the picturebox.

ScaleWidth

Type: Integer

Availability: Run-time Only

Remarks:

Returns the number of pixels for the horizontal measurement of the interior of the picture box. This is a read only control.

Example:

Width = AVPCPictureBox1.ScaleWidth

This will set the variable "Width" equal to the width of the picturebox.

Other Methods

CaptureScreen

Syntax:

AVPCPictureBox.CaptureScreen(xScrnLeft as Long, yScrnTop as Long, Width as Long, Height as Long)

Remarks:

Method captures an area of the desktop.

Example:

AVPCPictureBox1.CaptureScreen 0, 0, 255, 79

This will display an area of the desktop with the upper left coordinate (0,0) and lower right coordinate (255,79)

ClearSign

Syntax:

AVPCPictureBox.ClearSign

Remarks:

This method clears the marquee display.

Example:

AVPCPictureBox1.ClearSign

DrawPicture

Syntax:

AVPCPictureBox.DrawPicture(LoadPic_Path as String)

Remarks:

Method sets the path to a picture for the Picture Box picture property

Example:

AVPCPictureBox1.DrawPicture ("C:\pic\pic.bmp")

This will display a picture from the specified loaction.

SetDisplaySize

Syntax:

AVPCPictureBox.SetDisplaySize(SSizeX as integer, SSizeY as integer)

Remarks:

Method sets sign size. Must be set prior to transmission.

Example:

AVPCPictureBox1.SetDisplaySize 256, 128

This will set the display size to 256 pixels wide and 128 pixels high.

Other Properties

ConnectMode

Type: Byte [0 1]

0 – Serial Connection

1 – LAN Connection

Availability: Design-time or Run-time

Remarks:

Specifies the means for connecting to the marquee.

Example:

AVPCPictureBox1.ConnectMode = 1

This will set the program to use a LAN Connection.

DrawMode

Type: Integer: [1 - 16]

Availability: Design-time or Run-time

Remarks:

This property returns and sets a value that determines the appearance of output from a graphics method or the appearance of a **Shape** or **Line** control.

Example:

AVPCPictureBox1.DrawMode = 6

This will set the lines to draw in the inverse color of the background.

DrawStyle

Type: Integer: [0 - 6]

Availability: Design-time or Run-time

Remarks:

This property returns and sets a value that determines the line style for output from a graphics method.

Example:

AVPCPictureBox1.DrawStyle = 1

This will draw dash lines.

HasDC

Type: Boolean

Availability: Design-time or Run-time

Remarks:

Returns or sets a value that determines whether a unique display context (or hDC) is allocated to a control.

Example:

AVPCPictureBox1.HasDC = FALSE

No hDC is allocated to the control.

hDC

Type: Long

Availability: Design-time or Run-time

Remarks:

Returns a handle provided by the Microsoft Windows operating environment to the device context of an object.

Example:

Handle = AVPCPictureBox1.hDC

hWnd

Type: Long

Availability: Design-time or Run-time

Remarks:

Returns a handle to a form or control.

Example:

Handle = AVPCPictureBox1.hWnd

NetworkAddress

Type: String

Availability: Design-time or Run-time

Remarks:

Network address of sign. Can be an IP address, or network name, provided that the local computer can resolve it.

Example:

AVPCPictureBox1.NetworkAddress = "215.67.5.235"

This will set the network address as shown.

NetworkPort

Type: Integer: [0 – 32767]

Availability: Design-time or Run-time

Remarks:

Port at which to open socket connection with sign.

Example:

AVPCPictureBox1.NetworkPort = 2

This will set the socket connection to 2.

SerialComPort

Type: Byte: [1 - 16]

Availability: Design-time or Run-time

Remarks:

COM port to use for serial communication.

Example:

AVPCPictureBox1.SerialComPort = 2

This will set the Com port to 2.

ThresholdGreen

Type: Integer

Availability: Design-time or Run-time

Remarks:

Sets the minimum RGB green for sign.

Example:

AVPCPictureBox1.ThresholdGreen = 128

This will set the minimum RGB green to 128.

ThresholdInvert

Type: Boolean

Availability: Design-time or Run-time

Remarks:

Inverts the Red/Green thresholds.

Example:

AVPCPictureBox1.ThresholdInvert = FALSE

This will not invert the Red/Green thresholds.

ThresholdRed

Type: Integer

Availability: Design-time or Run-time

Remarks:

Sets the minimum RGB red for sign.

Example:

AVPCPictureBox1.ThresholdRed = 128

This will set the minimum RGB red to 128.

Alpha Pixel Protocol Version 1.0

1) Byte format

```
    bytes have the following format:

     B7 B6 B5 B4 B3 B2 B1 B0
    |  |  |  |  |  |  |  |  |
     \              /  ^  ^
      \            /   |  |
       \   DATA    /     \--\--  COMMAND CODE


COMMAND CODE is one of the following:
        B1  B0
        --  --
        0   0       Normal Pixel Data
        0   1       Compressed Pixel Data  (simple compression)
        1   0       End Of Line  (normal pixel data followed by start
  new line)
        1   1       Extended Command

Note: display is accessed via x,y coordinates (upper left is 0,0)
```

2) DATA format varies with the command code

```
COMMAND CODE            DATA means
----------------------  ------------------------------------------------
Normal Pixel Data        B7 B6 B5 B4 B3 B2
End Of Line             |  |  |  |  |  |  |
                         \   / \   / \   /
                         COLOR COLOR COLOR
                          Pm+2  Pm+1  Pm
(pixels displayed m through m+2, left to right, from current position)

Compressed Pixel Data    B7 B6 B5 B4 B3 B2
                        |  |  |  |  |  |  |
                         \         / \   /
                          REPEAT -4  COLOR
(3 pixels can be displayed with normal compression, so this allows 4 -> 19)

Extended Command         B7 B6 B5 B4 B3 B2
                        |  |  |  |  |  |  |
                         \               /
                          EXTENDED COMMAND


    COLOR bits are stored:  (MSBit, LSBit)

        00  BLACK
        01  RED
        10  GREEN
        11  YELLOW
```

EXTENDED COMMANDs: (B7 -> B2)

```
        (bit)        ----- (shift this value up 2 bit positions and OR in the Extended Comma
        765432      /       COMMAND CODE value, which is 0x03)
        ------     v
        000000  (0x00)      SOT     Start of Transmission
                                    (resets drawing pixel location to 0,0)
        000001  (0x01)      EOT     End of Transmission
                                    (signals transmission is complete)
    1 byte compression commands     (the next byte is a repeat count -20,
                                     as we can do up to 19 with the simple
                                     compression)
        000010  (0x02)      BLACK   Draw x+20 number of BLACK pixels
        000011  (0x03)      RED     Draw x+20 number of RED pixels
        000100  (0x04)      GREEN   Draw x+20 number of GREEN pixels
        000101  (0x05)      YELLOW  Draw x+20 number of YELLOW pixels

Future commands (unimplemented)
    2 byte compression commands     (the next byte is a repeat count -276,
                                     as we can do up to 275 with the single
                                     byte compression)
        000110  (0x06)      BLACK   Draw x+276 number of BLACK pixels
        000111  (0x07)      RED     Draw x+276 number of RED pixels
        001000  (0x08)      GREEN   Draw x+276 number of GREEN pixels
        001001  (0x09)      YELLOW  Draw x+276 number of YELLOW pixels
```

Cursor placement commands

```
        001010  (0x0a)      Start a new line  (zero x, increment y)
        001011  (0x0b)      Goto X, Y  (following two bytes)
        001100  (0x0c)      Goto XX, YY  (following four bytes)
```

Update window definition (this defines the update window; 'static' definitions remain once set until cleared and normal ones have to be set each transmission start; all cursor placement and draw commands will act in reference to the current set window, if any; current cursor position is reset to 0,0 each time one of these commands is issued; nothing yet to define multiple windows as there's nothing yet to define where the data is going) (added 04/03/02 sjh)

```
        010000  (0x10)  Set update window X1, Y1, X2, Y2  (following 4 bytes)
        010001  (0x11)  Set update window X1, Y1, X2, Y2  (following 4 bytes) (static)
        010010  (0x12)  Set update window XX1, YY1, XX2, YY2  (following 8 bytes)
        010011  (0x13)  Set update window XX1, YY1, XX2, YY2  (following 8 bytes) (static)
        010100  (0x14)  Clear current statically (or otherwise) set update window
```

Other

```
         100000  (0x20)      Perform factory service action XX  (2 bytes follow)  (added 04/0
02 sjh)
        100001  (0x21)      Enter factory services mode  (added 04/05/02 sjh)
        100010  (0x22)      Exit  factory services mode  (added 04/05/02 sjh)
        101000  (0x28)      Enter display DIM mode  (added 04/19/02 sjh)
        101001  (0x29)      Exit  display DIM mode  (added 04/19/02 sjh)
        101010  (0x2a)      Enter display direct RAM access mode  (added 04/19/02 sjh)
        101011  (0x2b)      Exit  display direct RAM access mode  (added 04/19/02 sjh)
        101100  (0x2c)      Perform display RAM page swap  (added 04/19/02 sjh)
        101101  (0x2d)      Blank display memory  (added 04/19/02 sjh)
```