

**EVALUATION KIT
OSD FOR MONITORS**

I - INTRODUCTION

I.1 - Demo Kit Contents

The STV942x OSD for monitors Demo Kit comes with the following parts :

- The STV942x Demo Board,
- The I²C Interface Board,
- One I²C Bus Cord,
- One VGA Cord,
- One 3.5" Floppy Containing the Page Editor Software,
- The Demo Kit User's Manual.

I.2 - Minimum Hardware and Software Configuration

The Page Editor Software is delivered on a MS DOS 3.5" floppy disk.

It requires a PC compatible microcomputer :

- Based on a 486 microprocessor or better,
- With at least 640k of memory.

Use of a 386 or smaller microprocessor will just give long execution time.

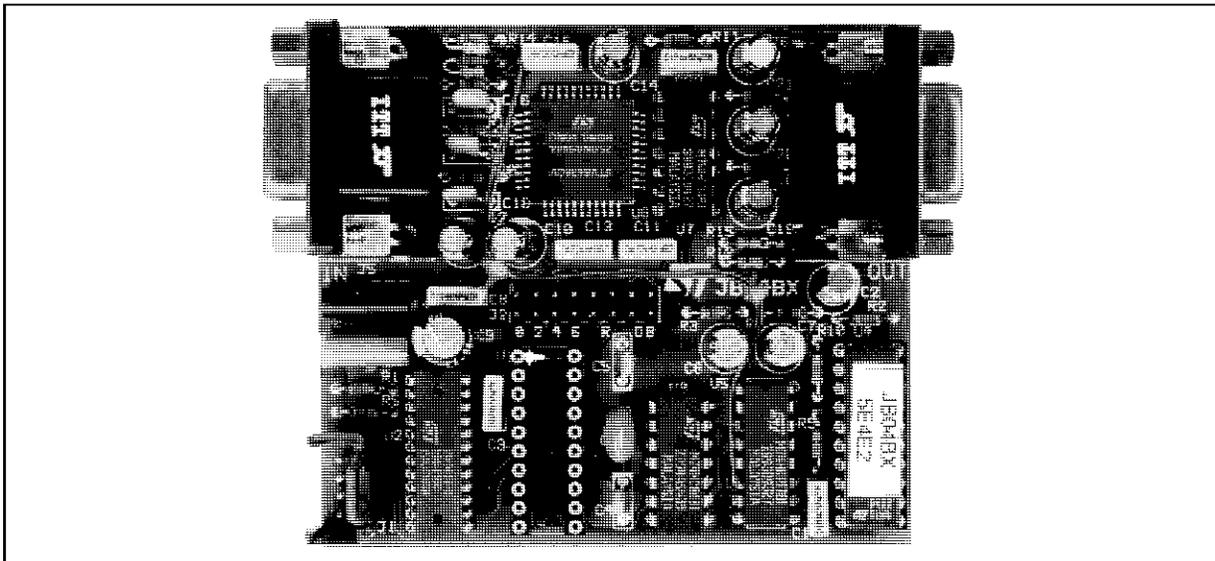
I.3 - How to Get Started

Even if it isn't necessary to connect the application board to run the Page Editor Software, it will always be helpful to see the real resulting page on the application monitor screen.

You may use the Demo Kit for an evaluation purpose, or for developing your OSD pages for your application. The STV942x Demo Board is supplied with VGA connectors and it comes with a VGA cord. The STV942x devices can handle a wide variety of monitors and video standards. If you are using the STV942x Demo Kit for evaluating the devices, or for developing a VGA monitor application, you will find it is perfectly suited for those purposes. If you use the kit to develop a non-VGA compatible monitor, you will find the different signals on the connectors and you will have to provide the appropriate connections to your application. Refer to the STV942x Demo Board Section.

Connect the I²C interface board on the parallel port LPT1 of your PC, then connect the I²C interface board to the STV942x Demo Board using the I²C bus wire, supply the Demo Board with a 5V power supply (not provided), and connect the Monitor to the Demo Board output connector J6, and the PC video output to the input connector J5, using the VGA cable provided with the Demo kit.

After having made a copy of the Page Editor Software under a working directory of you hard disk, you can start the execution by typing the command : **STV942X.exe**.



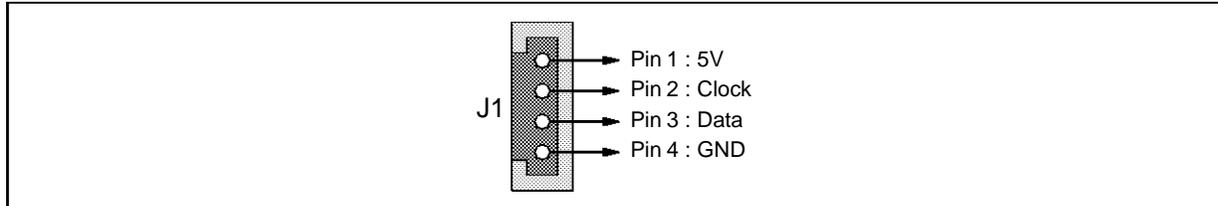
942X.TIF

STV942x - DEMO KIT

II - STV942x DEMO BOARD

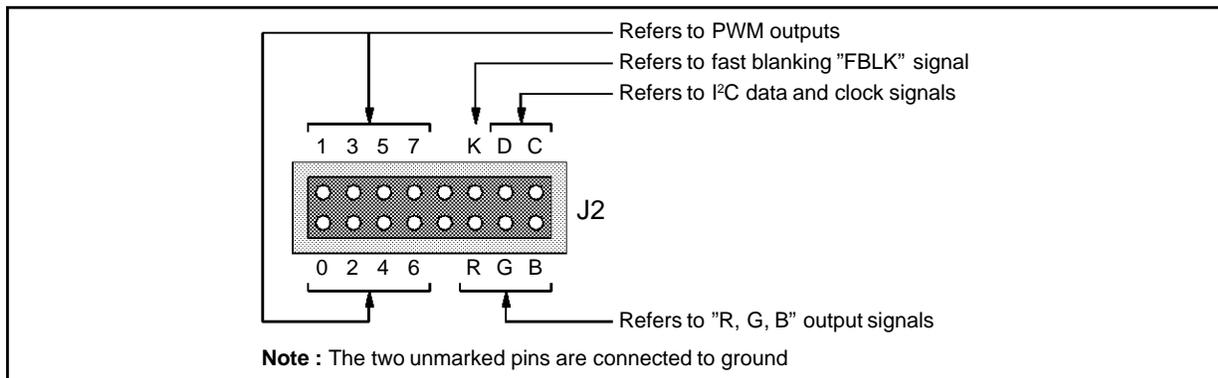
II.1 - Connectors

Figure 1 : J1 : I²C Connector



942X-01.EPS

Figure 2 : J2 : OSD Signals Test Points



942X-02.EPS

J3 : Power Supply +5V $\pm 5\%$

J5 : VGA, to Video Source

J4 : Power Supply Ground

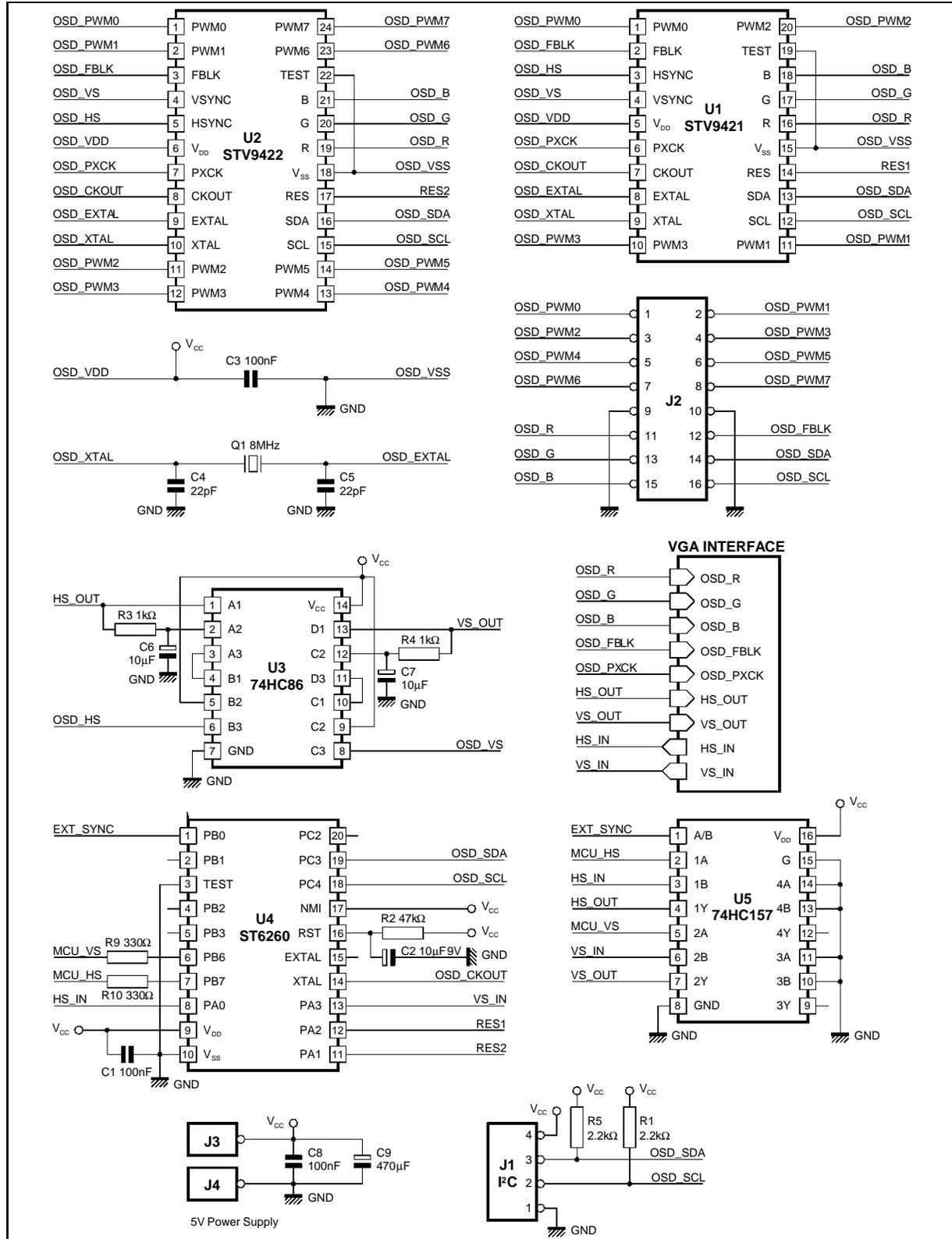
J6 : VGA, to Target Monitor

Table 1 : J5 and J6 Pin Description

Pin 1	Red	Pin 6	GND	Pin 11	NC
Pin 2	Green	Pin 7	GND	Pin 12	NC
Pin 3	Blue	Pin 8	GND	Pin 13	H SYNC
Pin 4	NC	Pin 9	NC	Pin 14	V SYNC
Pin 5	GND	Pin 10	GND	Pin 15	NC

II.2 - Circuit Diagram

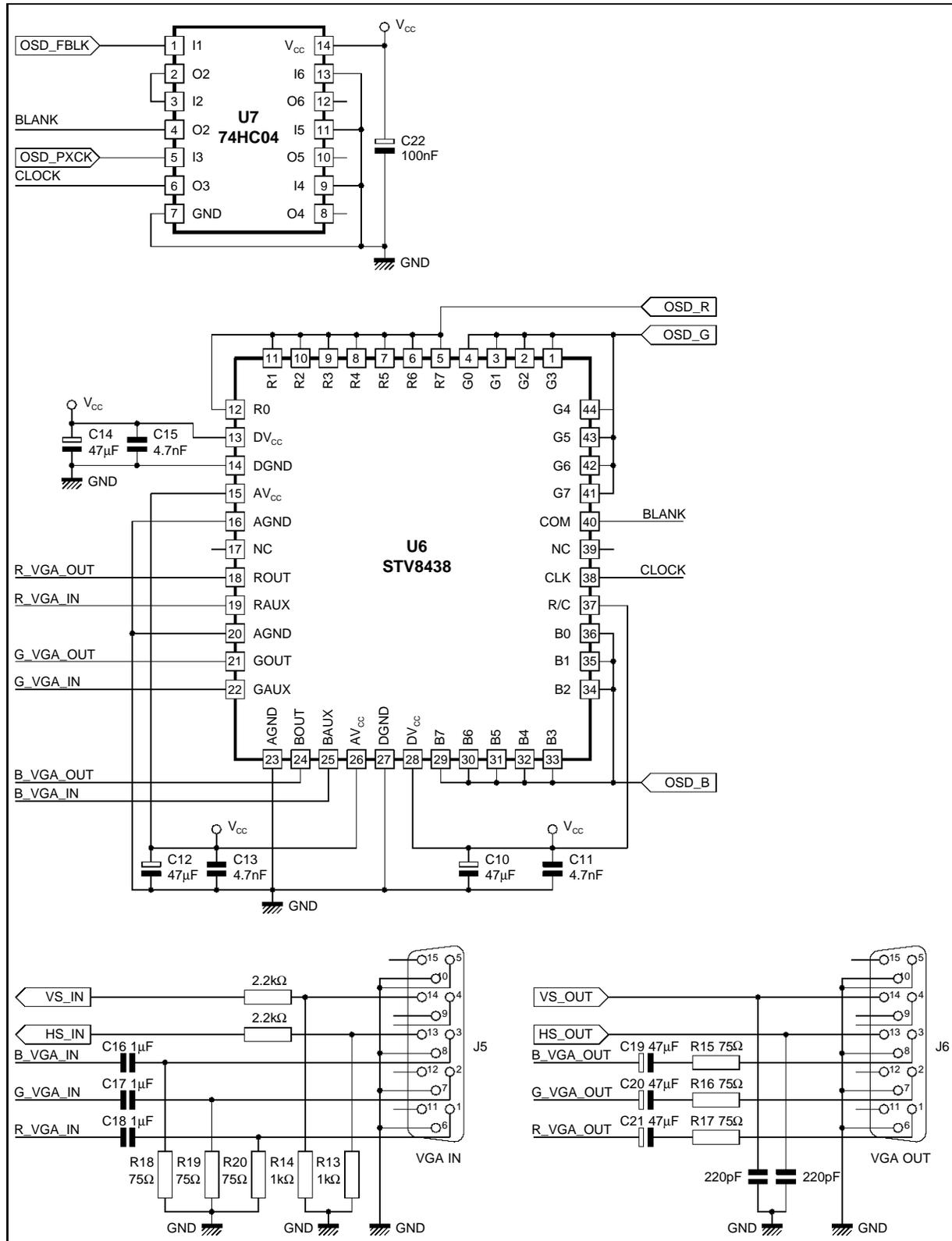
Figure 3



942X-03.EPS

STV942x - DEMO KIT

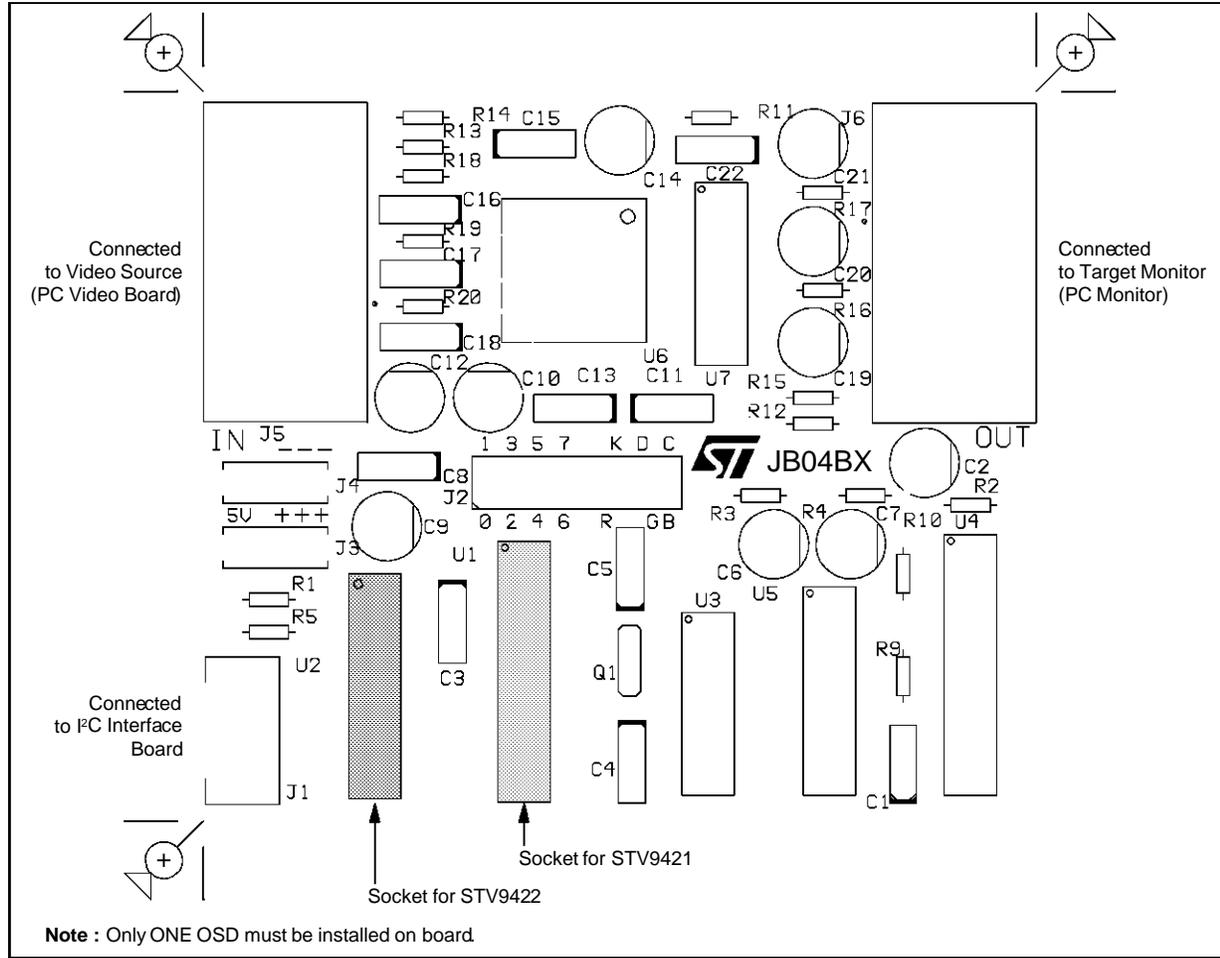
Figure 4



942X-04.EPS

II.3 - Board Layout

Figure 5



II.4 - Demo Board Operation

The STV942x Demo board is designed to be inserted into a VGA connection.

The video signals generated by the OSD device are mixed with the incoming video signals at the input connector J5. The combination of both signals is available on the output connector J6. The mixing is done with the **video switch/triple DAC : STV8438**. This device provides a triple analog video input, a triple digital video input, a video switch and three buffers for 75Ω drive. It operates at 100MHz on the analog input and on the video switch, and up to 70MHz on the digital inputs. In this application, the three 8 bits converters are used as 3 single bit converters as the OSD provides 1 bit RGB signals.

A **ST6 micro controller** gives the board the ability to run a small stand-alone demo program : It delivers two Horizontal and vertical synchronization signals, at about 31kHz and 60Hz, and drives the OSD I²C bus. As soon as an external sync is detected on the input VGA connector, the mcu stops delivering its own sync and it transmits the input sync signals to the VGA output connector. In a same manner, as soon as the mcu detects an external master on the I²C bus, it stops driving its demo program.

The Demo board can be equipped either with a STV9421 OSD (on socket U1), or with a STV9422 OSD (on socket U2) Never put both OSDs on the board at the same time !!

III - STV9420/21 PAGE EDITOR SOFTWARE

III.1 - Installation

In order to have a quick execution time, you should copy the page editor software from the 3.5" diskette to your hard disk.

It isn't absolutely necessary to connect the I²C interface, the demo board and the target monitor, to run the page editor software. If you run it alone, you can edit your application pages, without possibility to check the result on your application monitor screen. If you run it while the demo board and the target monitor are connected to the PC through the I²C interface board, the OSD device will be loaded at each command or character entered from the editor, so you will see the results in real time and in real size on your screen.

After having reset the demo board by pushing the RESET button, you start the execution by typing the command "**STV942X.exe**".

Depending on the LPT1 : parallel port address, an option must eventually be added on the command line :

- If LPT1 address is Hex 278 :
type "STV942X.exe /A",
- If LPT1 address is Hex 378 (default) :
type "STV942X.exe /B", or "STV942X.exe",
- If LPT1 address is Hex 3BC :
type "STV942X.exe /C".

III.2 - How to Build a Page

The OSD device can display up to 8 different pages stored in RAM. The number of the current displayed page is defined by three bits of the "Display Register".

That number "n" (0 to 7), refers to a RAM address (@ = n * 64), that must contain a list of descriptors : Vertical space descriptors and Text descriptors.

The Text descriptors contain the address of the lines of text.

Example : suppose you want to display the text "Hello, world", in page 3 :

1. Choose the page number : "F1" (edit registers), put "1" in bits 0 and 1 of the "Display ctl Register". In the RAM Zone, the yellow marker points the third line of RAM. Here, you must put the list of descriptors of the page.

2. This easy page is made of three strips : one top margin, one line of text, and one bottom margin. The three first locations of the RAM Line will contain : one Vert. Space Descriptor, one text descriptor, and one Vert. Space Descriptor : "F4", "F5", "F4". When you edit the Text Descriptor ("F5"), you are prompted to define the place of the text. You can put it anywhere, except in your descriptors list.
3. Type your text "Hello, World", at the location you defined before, when you edited the text descriptor ; terminate the text with "End of line" ("F7"). The text will be displayed on your target monitor. You can change the text color by changing the current color ("F2"), and repainting the text with the cursor : "CTRL Arrows".

III.3 - Description

The Page Editor software shows on just one screen the contents of the OSD device registers and RAM. The screen is divided in two parts :

Left Part : The OSD RAM

It is made of 16 lines of 32 positions, for a total of 512 positions. Each position represents 2 bytes of RAM and it can be edited with the OSD RAM Editor. When starting the OSD Page Editor, the RAM is blanked.

One position can hold :

- A character code : one byte for the code, one byte for the attributes,
- A text descriptor, containing, in two bytes, the address of one line of characters,
- A vertical space descriptor, containing, in two bytes, the number of space lines,
- One line of a user defined character, in two bytes. A complete user defined character takes 18 lines, or 36 bytes.

Right Part : The OSD Registers

It shows the contents of the 12 registers of the OSD device. Each register can be edited with the command "F1". When starting the OSD Page Editor, the registers are initialized with default values that are well suited for a VGA application.

During the OSD Page Editor Execution, when a RAM position, or a Register is edited, the OSD device is updated through the parallel port of the PC and the I²C interface.

III.4 - OSD RAM Editor

The OSD RAM Editor is running when the Page Editor Software starts.

The active keys are :

- **Standard Character keys :**

They correspond to the ASCII character codes Hex 20 to Hex 7F.

When depressing one of those keys, the corresponding character is put in current RAM position (cursor position), with the current color attributes (set by "F2" command). The RAM position is always overwritten.

Characters whose codes aren't in the above range are available with the "Special Codes" command "F6".

- **Arrow keys :**

The four arrow keys are used to move the cursor by one position, depending on the arrow direction. There is no change in the RAM contents, but at each move of the cursor, the content of the last position is sent to the OSD device. When the cursor is placed on a position containing a descriptor, a special code character, or a user defined character (on its first line), the detailed contents is displayed on an additional window.

- **CTRL, Arrow :**

This function is made to change the color of several characters without retyping them : When an Arrow key is depressed in conjunction with the CTRL key, the current position is rewritten with the current color attributes, then the cursor moves accordingly with the arrow direction. So, moving the cursor with CTRL, Arrow keys will repaint the underlying characters with the current color.

- **"TAB" :**

Action on this key toggles the "OSD Display" bit of the OSD Display control register (bit 7, @3FF3). The result of this is that in development configuration where the Demo board is inserted between the PC and the PC monitor, one can easily switch the monitor display between the Page Editor screen and the OSD output screen.

- **"Home" :** Moves the cursor to the beginning of the current line.
- **"End" :** Moves the cursor to the end of the current line.
- **"Page Up" :** Moves the cursor to the top line, on the current column.
- **"Page Dn." :** Moves the cursor to the bottom line, on the current column.
- **"Return" :** Moves the cursor to the beginning of next line.
- **"Del" :** Deletes the current position and shifts back rest of the line. Last position of the line is blank. This function operates only if there is no user defined

character stored between current position and end of line.

- **"CTRL, Del" :**

Same function than "Del" except that the current position content is stored into a temporary memory that can be recalled with "CTRL, Ins".

- **"Ins" :**

Inserts a blank into the current position. The rest of the line is shifted to the right. The last position of the line is lost. This function operates only if there is no user defined character stored between current position and end of line.

- **"CTRL, Ins" :**

Same than "Ins" except that the current position is not blanked, but written with the temporary memory content if already initialized by "CTRL, Del".

III.5 - Command "F1" : Registers Settings

To edit a register, depress the key "F1". A new window appears on the screen, that shows the detailed "bits" contents of the register currently under edition.

Active keys :

- **Down Arrow :** Edit the next register.
- **Up Arrow :** Edit the previous register.
- **Left Arrow :** Edit the bit on the Left.
- **Right Arrow :** Edit the bit on the Right.
- **0 or 1 :** Reset or Set the bit.
- **"Enter" :** Validation of the current bit and edit next bit.
- **"Escape" :** Exit from Registers Edition and go back to RAM Editor

III.6 - Command "F2" : Define Current Color

This command is used to define the current color attributes. Those attributes remain available until the next "F2" command. The current color is visible in the color window, under the RAM and Registers windows.

The color definition is made in two steps :

- **Foreground and Background colors definition :**

At the left side of the color window is a color selector, made of an upper palette for the choice of the foreground color, and a lower palette for the choice of the background color, including transparent background.

Up and Down Arrow keys move the selector to the upper palette and to the lower palette. *Left and Right Arrow* keys move the selector from color to color inside a palette. The resulting color is shown in a rectangle, at the right side of the palettes. The validation of the color is made by "RETURN" key. Depressing the RETURN key defines the new color as the current color and enters the second step of the definition.

- **Blinking, and shadowing attributes :**

The first attribute to define is the Blinking attribute. "SPACE" Bar sets or resets the blinking attribute. *Left and Right Arrows* move from selector to selector. "RETURN" validates the attribute and brings either to quit the color definition command, if the background color is non-transparent, or to define the shadowing attributes, if the background was defined as 'Transparent' in the previous step. "SPACE" Bar sets or resets the attributes and "RETURN" validates the choice.

At any point of the color definition, depressing "ESCAPE" key aborts the command without changing the current color.

III.7 - Command "F3" : User Character Define

This command is used to create or edit a user defined character in RAM. The character is made of 19 lines of 12 pixels, so it occupies a block of 38 bytes in the RAM space. There are 26 places in the RAM where to put user defined characters. The code corresponds to the block number : 0 to 25.

The command executes in two steps :

- **First step** : Choose Character place in RAM :
A special cursor, 19 positions long, is displayed in the RAM zone. It shows the code number of the user defined character that would occupy its place. *Arrow keys* move the cursor and allow to choose the place of the RAM character to be edited. The place will define the code number. "ESCAPE" key aborts the command. "ENTER" key validates the character place and enters the second step of the command.
- **Second step** : Edit the user defined character :
A new screen is displayed. It is made of three windows :
 - Bottom left window : it represents a character, made of 19 lines of 12 pixels.
Arrow keys move the edit cursor inside the character form.
"SPACE" bar lights on or off the current pixel. "RETURN" validates the character and exits the command. The 38 bytes of the user character will overwrite the block selected in the step one. "ESCAPE" aborts the command. "TAB" brings to the Upper left window.
 - Upper Left window : It allows to define the 6 characters that will be displayed on the Right side window. It provides a facility to edit multi-character symbols. The six fields corresponding to the six character codes are edited with the *numeric keys*. The default code is the non-existent character code 26 so that no default char-

acter is loaded in the right side window at the beginning. The *Left Arrow key* moves from field to field, "RETURN" validates the field and "ESCAPE" returns to the User character edition.

- Right side window : It is not editable and it is loaded with the user characters that have their codes initialized in the upper left window.

III.8 - Command "F4" : Vertical Space Descriptor

This command inserts a Vertical Space Descriptor at the cursor position. It is symbolized by a red double arrow in the RAM space. The user is prompted to give the number of space lines (1 to 255). The command can be aborted by "ESCAPE", and it is validated by "RETURN".

III.9 - Command "F5" : Text Descriptor

This command inserts a Text Descriptor at the current cursor position in the RAM space. It is symbolized by a red character "§". When entering the command, a new red cursor is displayed in the RAM window. It must point the text and the address will be stored in the descriptor. When the red cursor is placed, press "RETURN" to validate. A new "Text Strip Descriptor" window displays two selectors : "Dbl. Height" text and "Display" text. *Right and Left Arrows* move from selector to selector, "SPACE" bar sets or resets the selector, and "RETURN" key validates and exits the command. At any moment, the command can be aborted by "ESC" key.

III.10 - Command "F6" : Special Code

Inserts at the current cursor position a character with the current color, blinking and shadow attributes (defined by "F2" command). The character is defined by its code.

Two possibilities : either a character from the built-in ROM Font, or a user defined character (by "F3" command) stored in the RAM.

The command displays a "Special Code" window, with a selector : "User Code" and a data field "Code Number". Left and Right arrows move from the selector to the data field, "SPACE" bar sets or resets the selector, and the numeric keys are available to give the code number. For ROM font Characters, the available codes are in the range 0 to 127, but codes Hex 20 to 7F are reachable more efficiently with the alphanumeric keys under the Page Editor. For User defined characters, the codes are in the range 0 to 25 ; codes 26 to 127 are interpreted as "Spaces", and codes 64 to 127 are interpreted as End Of Line Characters.

III.11 - Command "F7" : End Of Line

This command inserts a character defined as "User" character with code 127, with the current color, blinking and shadow attributes. The result is a "Space" character that terminates the current strip of text : the following text in the RAM, if existing, is not displayed by the OSD in the same strip.

III.12 - Command "F8" : Load OSD File

Load registers and RAM from a file. The file must have been created before, using the "Save" command. The user is prompted to give a path and a file name.

III.13 - Command "F9" : Save OSD File

The Registers and RAM contents are saved in ASCII form in a file, together with a symbol table used by the page editor software. As no automatic save is done when exiting the page editor, the user should take care doing it before "Quit" command.

III.14 - Command "F10" : Quit

This command exits the Page Editor software without saving anything. Storage of RAM and Registers contents is done separately by the "Save" command (F9). The QUIT command is available only under the page editor. It is not possible to quit from an other command.

Information furnished is believed to be accurate and reliable. However, SGS-THOMSON Microelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No licence is granted by implication or otherwise under any patent or patent rights of SGS-THOMSON Microelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. SGS-THOMSON Microelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of SGS-THOMSON Microelectronics.

© 1995 SGS-THOMSON Microelectronics - All Rights Reserved

Purchase of I²C Components of SGS-THOMSON Microelectronics, conveys a license under the Philips I²C Patent. Rights to use these components in a I²C system, is granted provided that the system conforms to the I²C Standard Specifications as defined by Philips.

SGS-THOMSON Microelectronics GROUP OF COMPANIES

Australia - Brazil - China - France - Germany - Hong Kong - Italy - Japan - Korea - Malaysia - Malta - Morocco
The Netherlands - Singapore - Spain - Sweden - Switzerland - Taiwan - Thailand - United Kingdom - U.S.A.