

SENSORAY CO., INC.

---

**S622**  
**Software Development Kit**  
**version 1.0**

August 2003

© Sensoray 2003  
7313 SW Tech Center Dr.  
Tigard, OR 97223  
Phone 503.684.8073 • Fax 503.684.8164  
sales@sensoray.com  
www.sensoray.com



---

# Table of Contents

- INTRODUCTION ..... 3
- SOFTWARE INSTALLATION ..... 3
  - Windows98 ..... 3
  - WindowsNT ..... 3
  - Windows2000/XP ..... 3
  - Performance issues ..... 4
- BUILDING AN APPLICATION WITH S622.DLL ..... 4
- DATA TYPES REFERENCE ..... 5
  - System structure PCI ..... 5
  - Image buffer structure BUFFER ..... 6
  - Operation mode data structure MODE ..... 6
- FUNCTIONS REFERENCE ..... 7
  - S622\_InitSystem ..... 7
  - S622\_CloseSystem ..... 7
  - S622\_OpenBoard ..... 8
  - S622\_CloseBoard ..... 8
  - S622\_GetFreeBuffer ..... 8
  - S622\_QueueBfr ..... 9

# Introduction

Model 622 is a TV display board. It provides the capability to display images stored in JPEG format on a TV monitor.

The board supports the following JPEG file sizes: 640x480, 704x480, 320x240 and 352x240.

The SDK provided with the board allows building the application that displays a sequence of JPEG images with an arbitrary frame rate. A sample application that comes with the SDK illustrates the use of the SDK functions.

The general approach implemented in the SDK is that of a queue of internal buffers. A display application requests a buffer from the driver, fills it with JPEG data and puts it in the queue. The driver performs JPEG decompression and writes the uncompressed data to the board. In case the queue runs empty, the last frame is repeated, providing continuous video signal on the board's output.

## Software Installation

It is recommended to install the SDK before installing the board(s) into your PC. To install the SDK run setup.exe from the installation disk.

### Windows98

The first time you boot your system after model 622 frame grabber is plugged in, Windows will launch "Add new hardware Wizard". Select "Display a list of all drivers", click "Next", check "Floppy drive" and uncheck all other check boxes, and insert a S622 SDK distribution disk into the floppy drive. Click "Next". Windows should automatically complete the rest of the installation procedure.

### WindowsNT

To install the SDK run setup.exe from the installation disk.

### Windows2000/XP

The first time you boot your system after model 622 frame grabber is plugged in, Windows will launch "Found new hardware Wizard". Select "Install from a list...", click "Next", check "Include this location in the search", click on "Browse", and select a floppy drive. Insert a S622 SDK distribution disk into the floppy drive and click "OK". When a message appears saying that the driver does not have a Microsoft digital signature, click "Continue anyway".

**Note:** under WindowsXP, installing the driver before the SDK is installed may result in failure. In this case "Found new hardware..." message is not displayed any more. To install the driver

properly go to Control Panel – System – Hardware – Device manager, locate “Model 622 JPEG capture board” under “Sound, video and game controllers” or “Other devices”, right click on it, select “Properties”, select “Driver” tab, and click on “Update driver”. Follow the steps described above.

## Performance issues

It is recommended not to run any applications that transfer large amounts of data or access hard drive heavily concurrent with the model 622 application. In case the board is not capable of transferring the necessary amount of data from the system memory, image may get distorted.

# Building an application with s622.dll

## Files to be included in the project

The following files are distributed with the SDK:

- `s622.h` – contains data types and constants definitions;
- `s622f.h` – contains exported functions declarations;
- `s622app.c` – contains exported functions and helper functions definitions.

When building an application with `s622.dll`, it is necessary to include `s622app.c` in the project. Please, do not try the following: `#include "s622app.c"`, it is not going to work! Instead, use “Add files to project”.

### IMPORTANT:

If MFC is used, an option “Not using precompiled headers” must be set in the project settings for `s622app.c`.

All modules containing calls to the `s622.dll` functions must include `s622f.h`. Please refer to the sample source code for an example of building an application with S622 SDK.

# Data types reference

## System structure PCI

```
typedef struct {  
    DWORD boards;  
    DWORD PCISlot[SYS_BOARDS];  
} PCI;
```

The PCI structure contains information about the frame grabber boards identified by the system.

### *boards*

Number of supported boards identified by the system.

### *PCISlot*

Array of slot numbers.

The PCI structure is initialized by `S622_InitSystem` function. The system constant `SYS_BOARDS` determines the maximum number of frame grabbers supported, and is defined in `s622.h`. The *boards* member is set by `S622_InitSystem` to the number of model 622 boards detected in the system.

*PCISlot* member contains PCI bus (upper 16 bits) and PCI slot (lower 16 bits) numbers for a given board. The PCI slot number is generated by BIOS and/or Windows, and usually is not the same as the ordinal number of the slot. On different systems, the order in which the boards are enumerated (from the CPU outwards, or reverse) may be different. The PCI bus/slot number is provided for reference only.

All SDK functions that need to access an individual board would address it with the help of *index* parameter, which is an index of a selected board in *PCISlot* array.

### Example:

```
PCI  pci;          //PCI struct  
ECODE ecode;      //error code  
int  i;           //board index  
  
if (!(ecode = S622_InitSystem(&pci)) {  
    for (i = 0; i < pci.boards; i++) {  
        ecode = S622_OpenBoard (i, Callback, NULL);  
        . . . . .  
    }}
```

## Image buffer structure BUFFER

```
typedef struct {  
    void *pJpeg;          //pointer to compressed data buffer;  
    DWORD size;          //actual size of jpeg data;  
    int reserved;        //reserved  
} BUFFER;
```

### *pJpeg*

Pointer to JPEG data. An application fills in the buffer with JPEG data.

### *size*

Actual size of the JPEG file image in the buffer.

### *reserved*

Reserved. Do not modify

## Operation mode data structure MODE

```
typedef struct {  
    DWORD format;          //NTSC, PAL  
    DWORD bmpsize;        //bitmap size  
} MODE;
```

The MODE structure contains display parameters.

### *format*

Defines the format of the output video signal:

FORMAT\_NTSC – NTSC (60Hz) video;

FORMAT\_PAL – PAL or SECAM (50Hz) video. (Not supported in v.1.0).

### *bmpsize*

Image size (has to match that of the JPEG image precisely):

SIZE0 – 640x480;

SIZE1 – 704x480;

SIZE2 – 320x240;

SIZE3 – 352x240.

# Functions reference

## S622\_InitSystem

```
ECODE S622_InitSystem (  
    PCI *pPci          //pointer to a PCI structure  
);
```

### Parameters

*pPci*  
Pointer to the structure of PCI type.

### Return values

Returns 0 in case of success, or an error code (a list of error codes is included in s622.h).

### Notes

The function initializes the driver, searches for all boards supported by the SDK. This function has to be called **only once** per application. The system resources allocated by S622\_InitSystem are released by a call to S622\_CloseSystem. See the description of the PCI structure for details.

## S622\_CloseSystem

```
void S622_CloseSystem (  
    void  
)
```

### Parameters

None.

### Return values

None.

### Notes

S622\_CloseSystem releases all resources allocated by the calls to S622 functions.

## S622\_OpenBoard

```
ECODE S622_OpenBoard (  
    int index,                //board index  
    MODE *pMode              //pointer to MODE  
)
```

### Parameters

*index*

A value by which the boards are addressed. The board selected by *index* has the PCI slot value of PCI.PCIslot[*index*]. The value of *index* has to be between 0 and PCI.boards - 1.

*pMode*

A pointer to a MODE type variable that has to be filled in before a call to S622\_OpenBoard.

### Return values

Returns 0 in case of success, or an error code.

## S622\_CloseBoard

```
void S622_CloseBoard (  
    int index                //board index  
)
```

### Parameters

*index*

A value by which the boards are addressed. The board selected by *index* has the PCI slot value of PCI.PCIslot[*index*]. The value of *index* has to be between 0 and PCI.boards - 1.

### Return values

None.

### Note

A call to S622\_CloseBoard disconnects the application from the selected board. Other applications are able to access the board after that. A call to S622\_CloseSystem closes all open boards automatically.

## S622\_GetFreeBuffer

```
BUFFER * S622_GetFreeBuffer (  
    int index                //board index  
)
```

### Parameters

*index*  
Board index.

### **Return values**

Returns a pointer to a free buffer (a structure of BUFFER type), or NULL if no buffers are available. The application has to call S622\_GetFreeBuffer until a pointer to a valid buffer is received.

### **S622\_QueueBfr**

```
void S622_QueueBfr (  
    int index          //board index  
    BUFFER *pBfr      //pointer to the BUFFER structure  
)
```

### **Parameters**

*index*  
Board index.

*pBfr*  
Address of the BUFFER structure obtained with S622\_GetFreeBuffer.

### **Return values**

None.