```
/*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++

$Id: rgbvideo.c,v 1.9 2001/10/02 21:12:54 guyc Exp $

Copyright (c) 2001 BeComm Corporation

Filename:

    rgbvideo.c

Abstract:

    Displays rgb video to the default rgbdisplay device.

Owner:

    Guy Carpenter (guyc)

------------------------------------------------------------------------
---*/

#define SOS_DEBUG_ZONE "/beads/rgbvideo"

#include <sosstrings.h>
#include <sosmultimedia.h>

SOS_SOURCE_VERSION("$Id: rgbvideo.c,v 1.9 2001/10/02 21:12:54 guyc Exp
$");

/*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++
Named Constants
------------------------------------------------------------------------
---*/

/*
 * Name of bead
 */
static const char BEAD_NAME[] = "rgbvideo";

// REVISIT - device name should not be hard coded.
/*
 * Name of the device we use for output.
```

```
41 #define DEVICE_CLASS "rgbdisplay"
42 #define DEVICE_NAME  "default"
43
44 /*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
…  +++++
45 Structs
46 ------------------------------------------------------------------------------
…  ---*/
47
48 typedef struct {
49     char *                DeviceName;
50     void *                FrameBuffer;
51     size_t                FrameBufferSize;
52     SOS_VIDEO_FORMAT      VideoFormat;
53     SOS_IRGBDISPLAY *     IRgbDisplay;
54     SOS_ISAMPLECLOCK *    IRenderClock;
55     SOS_ISAMPLECLOCK *    IMasterClock;
56     SOS_IVIDEOCONTEXT *   IVideoContext;
57     SOS_VIDEO_TIMESTAMP   FrameNumber;
58     SOS_BOOLEAN           DisplayReady;
59     SOS_BOOLEAN           ClocksReady;
60 } RGBVIDEO_CONTEXT;
61
62
63 /*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
…  +++++
64 Context Stuff
65 ------------------------------------------------------------------------------
…  ---*/
66
67 static
68 void
69 RgbVideo_ContextDestroy(
70     RGBVIDEO_CONTEXT *        Context
71 )
72 {
73     if (Context) {
74
75         if (Context->FrameBuffer) {
76             SOS_Mem_Free(Context->FrameBuffer);
77         }
78
79         if (Context->IRgbDisplay) {
80             SOS_RegObject_InterfaceRelease(Context->IRgbDisplay);
```

```
82
83          if (Context->DeviceName) {
84              SOS_Mem_Free(Context->DeviceName);
85          }
86
87          if (Context->IMasterClock) {
88              SOS_Interface_Release(Context->IMasterClock);
89          }
90
91          if (Context->IRenderClock) {
92              SOS_Interface_Release(Context->IRenderClock);
93          }
94
95          if (Context->IVideoContext) {
96              SOS_Interface_Release(Context->IVideoContext);
97          }
98
99          SOS_Mem_Free(Context);
100     }
101 }
102
103 /*
104  * Find the device and get the necessary interface
105  */
106 static
107 SOS_STATUS
108 RgbVideo_DeviceInit(
109     RGBVIDEO_CONTEXT *      Context
110 )
111 {
112     SOS_STATUS status;
113
114     SOS_REGOBJECT *device;
115
116 /*
117  * NOTE : if InterfaceGet fails it will set IRgbDisplay
118  * to NULL for us.
119  */
120     status = SOS_Registry_DeviceGet(
121         DEVICE_CLASS,
122         Context->DeviceName,
123         &device);
124
125     if (SOS_SUCCEEDED(status)) {
```

```
127             device,
128             SOS_IRGBDISPLAY_ID,
129             (void**)&Context->IRgbDisplay
130         );
131
132     }
133     SOS_RegObject_Release(device);
134
135     return status;
136 }
137
138 static
139 RGBVIDEO_CONTEXT *
140 RgbVideo_ContextCreate(
141     void
142 )
143 {
144     SOS_STATUS status = SOS_Success;
145     RGBVIDEO_CONTEXT *context;
146
147     context = SOS_Mem_Alloc(sizeof(*context));
148     if (context) {
149         SOS_memset(context, 0, sizeof(*context));
150
151         /*
152          * Revisit - this should not be hard coded
153          */
154         context->DeviceName = SOS_strdup(DEVICE_NAME);
155
156         status = RgbVideo_DeviceInit(context);
157
158     } else {
159         /* failed mallocing context */
160         status = SOS_ErrorResourceAllocation;
161     }
162
163     /*
164      * If anything went wrong we should cleanup
165      */
166     if (SOS_FAILED(status)) {
167         RgbVideo_ContextDestroy(context);
168         context = NULL;
169     }
170
171     return context;
```

```
172 }
173
174
175 /*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
… +++++
176
177 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
… +++*/
178 static
179 SOS_STATUS
180 DisplayPrepare(
181     RGBVIDEO_CONTEXT *      Context,
182     SOS_MESSAGE *           Message
183 )
184 {
185     SOS_STATUS status = SOS_Success;
186
187     status = Context->IVideoContext->FormatGet(
188         Context->IVideoContext,
189         Message,
190         &Context->VideoFormat,
191         NULL /* don't care about timestamp */
192     );
193
194     if (SOS_SUCCEEDED(status)) {
195         status = Context->IRgbDisplay->Open(
196             Context->IRgbDisplay,
197             Context->VideoFormat.Width,
198             Context->VideoFormat.Height,
199             SOS_VIDEO_RGB24
200         );
201     }
202
203     return status;
204 }
205
206 SOS_STATUS
207 ContextClockGet(
208     SOS_PATH *              Path,
209     const char *            Name,
210     SOS_ISAMPLECLOCK **     ISampleClock
211 )
212 {
213     SOS_STATUS status;
    SOS_RECORJECT *object;
```

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.