good", "yellow" to indicate "video standby", and "red" to indicate "no video controller power." All of the LEDs can be programmed to blink or flash when an inverter failure is detected or an inverter **202/204** is shut off by the controller **500** due to an over-temperature condition. The LEDs indicate an overall status and more detailed data can be recorded in log contained in the memory associated with the controller **500**.

[**0046**] **FIG. 6** is a block diagram of the data flow in an exemplary backlit LCD monitor control system. The controller **500** receives data that could include fan current data from the fans **402/408**, log data from the memory **602** associated with the controller **500**, data from the I/O device **510**, data from the video controller **512**, user input brightness control **508** data, brightness data from the brightness sensors **506**, temperature data from the temperature sensors **502**, and inverter current data from the inverters **202**. The controller **500** can use this data to send instructions to the fans **402** to specify and set the speeds of the fans. The controller **500** can also use the data it receives to set the brightness of the bulbs **102** by setting the dimming control **604** which in response will turn the inverter **202/204** current up or down. The controller **500** can also use the data it receives to determine when and what to write to the log **602**, to determine data to send to the I/O device **510**, to determine how the lights on the LED indicators **504** should be set, and to turn the display on or off **606**.

[**0047**] **FIG. 7A** is a flowchart of an exemplary control process that can be executed by the controller **500**. The flowchart illustrates that the controller **500** monitors the status of several elements (e.g., bulbs, inverters) and stores a record of the status and of any failures on the log. The log records can then be downloaded to a personal computer, or other external device or communicated, real-time, to a monitoring computer for predictive fault analysis. Operators can also be alerted to faults through the LEDs **604** on the backlit LCD monitor. The process begins at step **702** when the power to the monitor is turned on. Next, at step **704** system initialization occurs which can include steps such as setting the inverters **202/204** and displaying an initial screen. Next, a loop is started at step **706** that goes through step **728** and is repeated until the monitor is powered off. At step **706**, the runtime counters are updated. The first time the loop is executed the runtime counters are started and with each subsequent execution of the loop the runtime counters are updated.

[**0048**] At step **708** brightness control is performed. Step **708** can be performed using a process such as that shown in **FIG. 7B**. The process depicted in **FIG. 7B** begins by determining whether a remote calibration box, a local potentiometer, or an override should be used to calculate inverter output. Once this output has been sent to the inverters, a brightness sensor failure recognition process, such as the one depicted in **FIG. 7C** is performed. As shown in **FIG. 7C** a brightness sensor event is stored in memory if it is time to check for brightness sensor failures and twelve or more sensors have failed. Processing continues back on **FIG. 7B** by performing a bulb decay process, such as the one shown in **FIG.** 7D, and a bulb decay compensation process, such as the one shown in **FIG.** 7E, if no brightness sensor failures have occurred. The bulb decay process shown in **FIG. 7D** includes storing a bulb decay event in memory and setting a new bulb decay break point in the event that bulb decay if has occurred. The bulb decay compensation process shown in **FIG. 7E** includes adjusting the bulb brightness control up to an original limit (here 1200 nits) and then letting the bulbs decay once the original limit has been reached.

[**0049**] Next, step **710** is executed to perform inverter monitoring for the first inverter. Step **710** can be performed using a process such as that shown in **FIG. 7F**. The inverter monitoring process checks for an open circuit, for over-current and for under-current conditions. If an open circuit condition exists and the inverter is not currently in an open circuit state, the failure event if logged and the power LED is set to blink. If the inverter is in an over-current state then the failure is recorded, the inverter is turned off and the power LED is set to blink. Similarly, when an under-current state exists, the failure is recorded, the inverter is turned off and the power LED is set to blink. Processing then continues with step **712** in **FIG. 7A**.

[**0050**] Step **712** is executed to perform inverter monitoring for the second inverter. The same process described in reference to step **710** is performed. If there are more than two inverters this inverter monitoring process is performed for each subsequent inverter. Next, at step **714** it is determined whether the PC, or I/O device **510** is requesting a download. If a download is being requested, step **716** is performed and the data from the event log is downloaded to the PC or I/O device **510**.

[**0051**] Next, step **718** is performed to monitor the system temperature. Data from the temperature sensors **502** is collected by the controller **500**. At step **720** the thermal control process is performed. Step **720** could be performed using a process such as the one depicted in **FIG. 7G**. As shown in **FIG.** 7G, the thermal control flow process includes checking for a critical temperature failure and if one exists, recording the critical temperature failure event and setting all three LEDs to blink. The actual temperature is then determined and if it is greater than normal, the fan speed is set. The temperature is then compared to various programmable temperature levels to determine if a caution temperature, over-temperature, phase 1 temperature, phase 2 temperature, or critical temperature failure condition exists. If the temperature is at a caution temperature level, the yellow temperature LED is lit. If the temperature is at an over-temperature, the event is recorded, a counter is started, and the red temperature LED is lit. Phase 1 temperature and phase 2 temperature readings result in recording the event, reducing the bulb brightness level and setting the red temperature LED to blink.

[**0052**] Next, step **722** is performed to monitor and control the fans **402/408**. The fan speed can be changed in response to data collected by the temperature sensors **502** or in response to data received by the I/O device **510**. A check is made at step **724** to determine if all the fans **402/408** are working properly. If they are not working properly, step **726** is performed to display the failure indicators on the LED and to record the failure in the event log. The backup system data is collected at step **728**. The loop begins again at step **706** and continues until the monitor is powered off.

[**0053**] **FIG. 8** is a flowchart of the data flow in an exemplary bulb brightness control module executed by the controller **500**. As shown in **FIG. 8**, the controller **500** executes a program that receives information from temperature sensors **502** and brightness controls **508**. The controller