

# ***Using the Watchdog Documentation Addendum***

*December 21, 2005*

## **WARNING**

EURESYS S.A. shall retain all rights, title and interest in the hardware or the software, documentation and trademarks of EURESYS S.A.

All the names of companies and products mentioned in the documentation may be the trademarks of their respective owners.

The licensing, use, leasing, loaning, translation, reproduction, copying or modification of the hardware or the software, marks or documentation of EURESYS S.A. contained in this documentation, is not allowed without prior notice.

EURESYS S.A. may modify the product specifications or change the information given in this documentation at any time, in its discretion, and without prior notice.

EURESYS S.A. shall not be liable for any loss of or damage to revenues, profits, goodwill, data, information systems or other special, incidental, indirect, consequential or punitive damages of any kind arising in connection with the use of the hardware or the software of EURESYS S.A. or resulting of omissions or errors in this documentation

## Contents

<b>1. Scope of this application note</b> .....	<b>3</b>
<b>2. What is a watchdog?</b> .....	<b>4</b>
2.1. General principle .....	4
2.2. Detailed operation .....	4
<i>Block diagram</i> .....	4
<i>Timing diagram</i> .....	7
<b>3. How to use the watchdog</b> .....	<b>9</b>
3.1. Hardware connection .....	9
3.2. Software application .....	10
<i>MultiCam API</i> .....	10
<i>Typical procedure</i> .....	10
3.3. Hints .....	11
<i>Watchdog resets logging</i> .....	11
<i>Partial watchdog usage</i> .....	11
<i>Remote signaling</i> .....	11
<i>Dealing with exceptional long boot</i> .....	11
<i>Minimum watchdog timeout</i> .....	12
3.4. Watchdog control parameters summary .....	12
<b>4. Glossary</b> .....	<b>13</b>

## **1. Scope of this application note**

Video surveillance systems based on PC are sophisticated by nature. As everybody knows, it is impossible to absolutely guarantee a reliable operation of a PC during several years.

Mains electrical perturbations can, for example, perturb the PC operation and lead to a failure of the video surveillance system. Human intervention is therefore required to restart it.

A PC equipped with Euresys frame grabbers is improved so that the video surveillance task is continuously guaranteed. This performance is accomplished by a hardware device, called watchdog, implemented on the Euresys frame grabber.

The purpose of this application note is to explain how to install and to configure the watchdog with only a few lines of program code.

## 2. What is a watchdog?

### 2.1. General principle

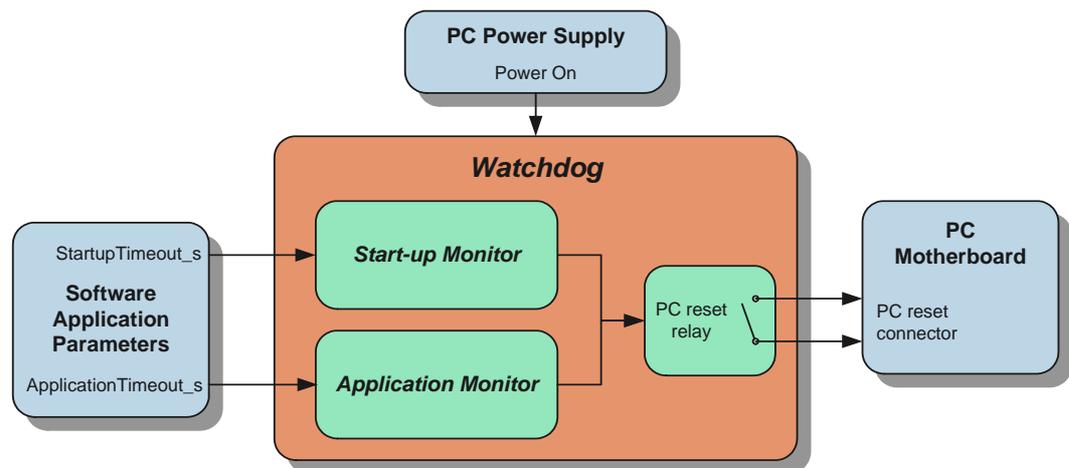
The watchdog is a hardware device which monitors the video surveillance software application running on the PC. If there is no activity of the software application during a defined time called “timeout”, the watchdog concludes in a system failure and closes a PC reset relay which restarts the system.

The watchdog parameters can be configured by the software application with the unified programming interface provided with Euresys frame grabbers and called MultiCam.

### 2.2. Detailed operation

#### Block diagram

The block diagram of the watchdog is shown here below.



The watchdog is build around two main devices called monitors:

- The **Start-up Monitor** is responsible for monitoring the system when the PC is booting and until the software application is launched.
- The **Application Monitor** is responsible for monitoring the system during the normal operation of the software application.

The monitors are based on a timeout mechanism. The software application must signal to the monitor that it is alive before a timeout period. If the application does not give any sign of life during this period, the timeout expires and the monitor concludes in a system malfunction.

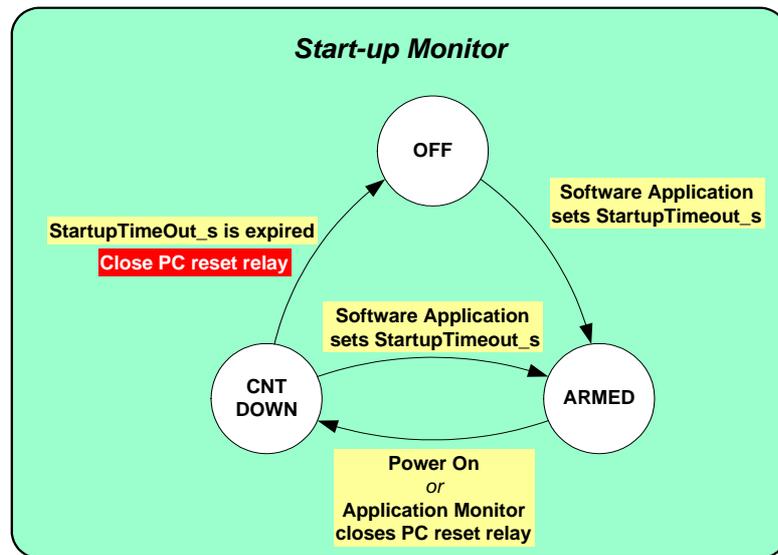
The implementation of two distinct monitors allows the programmer to define separate watchdog timeouts, one for the start-up phase and one for the normal operation phase.

The above block diagram connects with three inputs and one output:

- **Power On:** the watchdog requires the power supply of the monitored PC. At Power On, the watchdog is operational and starts to monitor the system.
- **StartupTimeout\_s:** the software application is responsible to adequately define a maximum time. It allows the system to complete the start-up sequence by setting the `StartupTimeout_s` parameter.
- **ApplicationTimeout\_s:** the software application is responsible to adequately define a maximum time. It allows the application to periodically signal its correct operation by setting the `ApplicationTimeout_s` parameter.
- **PC reset connector:** when the watchdog concludes that the start-up has failed or that the application is no longer running, it restarts the system by closing the PC reset relay. For this purpose, an electrical connection has to be installed between the PC reset relay and the reset connector of the PC motherboard.

### Start-up Monitor

The Start-up Monitor can be graphically represented by a state-device as follows:



The Start-up Monitor can be in one of the following three states:

- The Start-up Monitor is disabled in the OFF state.
- In the ARMED state, the start-up monitor is waiting for a start-up sequence to monitor.
- During the CNT DOWN state, timeout expiration can occur if the start-up sequence is too long.

The Start-up Monitor is able to memorize its state and the `StartupTimeout_s` parameter value. It is therefore fully operational as soon as power is applied and before the launch of the software application.

The factory preset of the Start-up Monitor is the OFF state. When the software application sets the `StartupTimeout_s` parameter, the Start-up Monitor goes in the ARMED state. In the ARMED state, the Start-up Monitor waits for a new start-up sequence.

The Start-up Monitor knows that a new start-up sequence is engaged when power is applied (Power On). Alternately, a start-up sequence can be engaged following a PC reset relay closure generated by the Application Monitor.

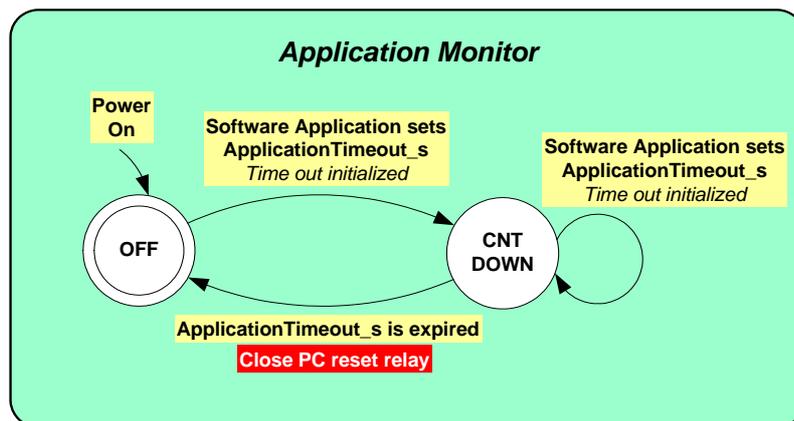
The Start-up Monitor goes in the CNT DOWN state and begins to count the time that elapses. Two situations may arise in the CNT DOWN state:

- The application is alive and **sets** the `StartupTimeout_s` parameter before the maximum timeout expiration and the Start-up Monitor concludes that the start-up sequence has been correctly completed. The Start-up Monitor goes in the ARMED state and is ready to monitor a new start-up sequence.
- The start-up is abnormally long and the application has **no chance to set** the `StartupTimeout_s` parameter before timeout expiration. The Start-up Monitor concludes that the start-up sequence has not been correctly completed. The Start-up Monitor goes in the OFF state and closes the PC reset relay for about 1 second.

Once the software application has successfully parked the Start-up Monitor in the ARMED state, the Application Monitor can be exercised.

### Application Monitor

The Application Monitor can also be graphically represented by a state-device as follows:



The Start-up Monitor can be in two states:

- The Application monitor is disabled in the OFF state.
- During the CNT DOWN state, timeout expiration can occur if the software application hangs-up and does not signal its activity.

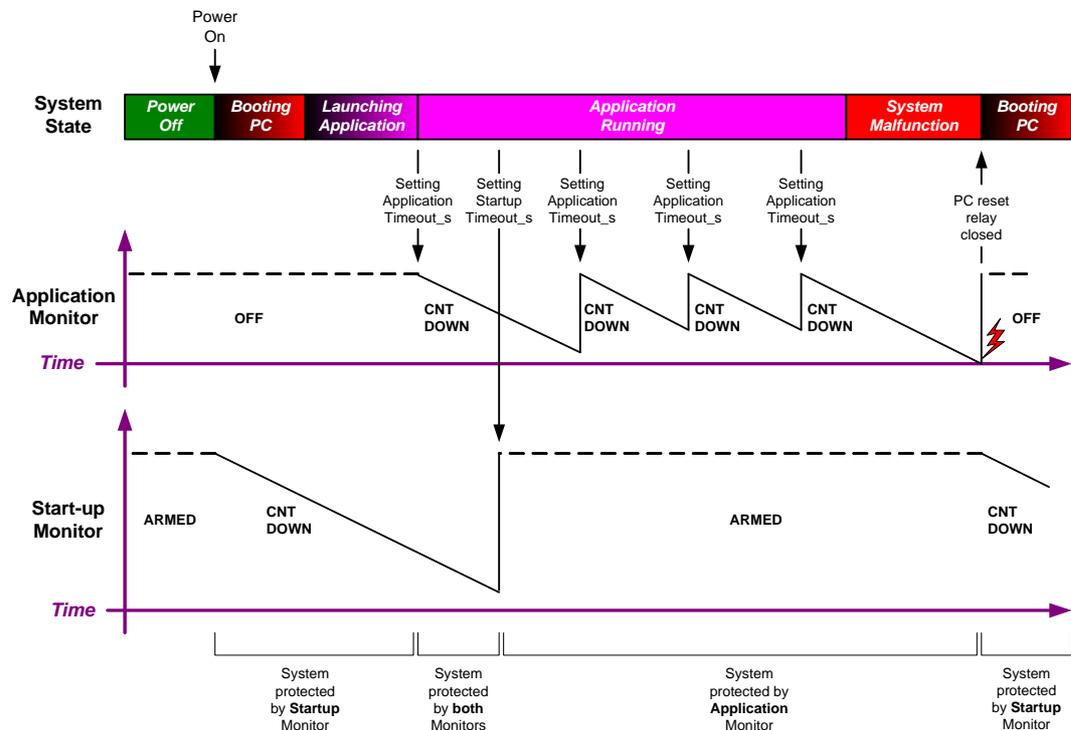
The Application Monitor has no means to memorize its state when power is switched off. At Power On, it always starts in the OFF state. The Application Monitor becomes active and goes in the CNT DOWN state under the software application control by simply setting the `ApplicationTimeout_s` parameter to a desired value.

Once in the CNT DOWN state, the Application Monitor waits for a new setting of the `ApplicationTimeout_s` parameter. Two situations may arise in the CNT DOWN state:

- The software application is alive and **sets** the `ApplicationTimeout_s` parameter before the timeout expiration. The Application Monitor concludes that the application is correctly running. The Application Monitor stays in the CNT DOWN state and waits for the next `ApplicationTimeout_s` parameter setting during the timeout period.
- A malfunction occurs and the application **does not run the appropriate code which sets** the `ApplicationTimeout_s` parameter. Timeout expiration occurs and the Application Monitor concludes that the application is no more running. The Application Monitor goes in the OFF state and closes the PC reset relay during about 1 second.

### Timing diagram

The timing diagram of the watchdog operation is shown below.



When the PC power supply is switched on, the Start-up Monitor goes in the CNT DOWN state. It is assumed that the software application has set the `StartupTimeout_s` parameter during a previous PC operation. The Start-up Monitor has therefore memorized an ARMED state.

After PC booting, the application is launched and begins its normal operation. The Application Monitor goes in the CNT DOWN state when the application sets the `ApplicationTimeout_s` parameter.

The Start-up Monitor counting down is stopped when the application sets the `StartupTimeout_s` parameter. At this point, the Start-up Monitor knows that the start-up has been successfully completed and that the application is running. The Start-up Monitor goes in the ARMED state and is ready to monitor the next start-up sequence.

The system is now in normal operation and the application periodically sets the `ApplicationTimeout_s` parameter to advise the Application Monitor that everything is working properly.

When a system malfunction occurs, the application is frozen and the internal counter of the Application Monitor reaches 0 after the timeout period.

The Application Monitor closes the PC reset relay during 1 second and goes in the OFF state. The relay closure generates a reboot of the PC. The Start-up Monitor goes from the ARMED state to the CNT DOWN state to monitor the coming start-up sequence.

To ensure system monitoring without interruption, the software application must perform a first setting of `ApplicationTimeout_s` before `StartupTimeout_s`.

## 3. How to use the watchdog

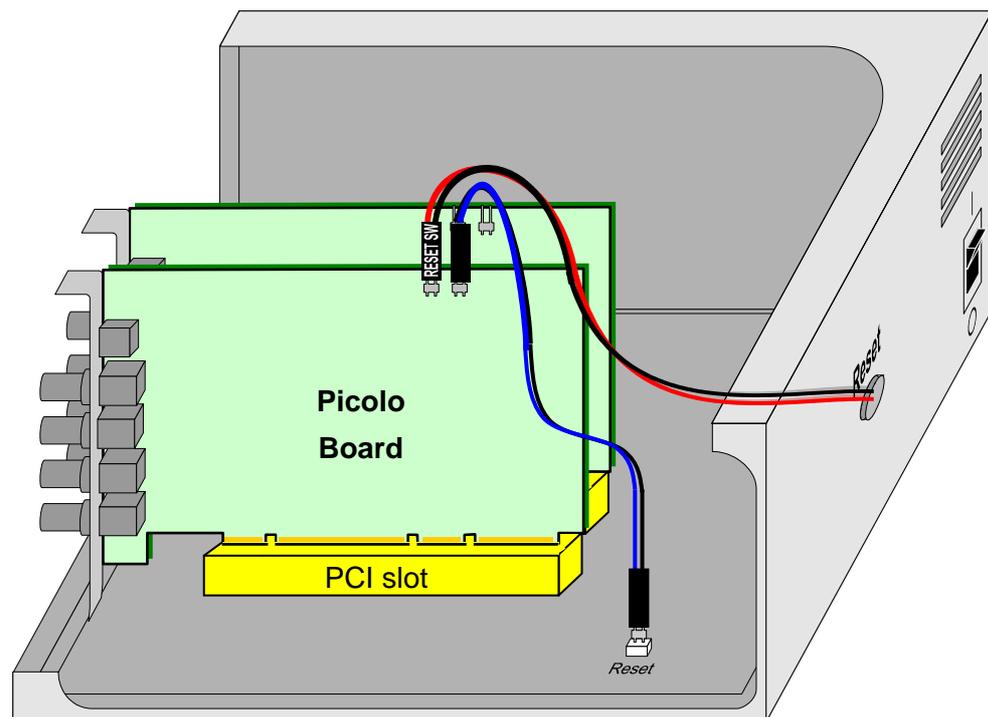
### 3.1. Hardware connection

The watchdog closes the PC reset relay during 1 second when it detects a malfunction of the system.

An electrical connection has to be installed between the PC reset relay and the PC reset connection on the motherboard. For this purpose, the Euresys frame grabbers are fitted with a PC reset header.

For the localization of the PC reset header, refer to the “*Euresys Hardware Documentation*”, select the “*Board family*”, go in the “*Installation guide*” and select the “*Product description*” and then “*Product appearance*”. A picture helps to locate the PC reset headers on the selected board.

If several frame grabbers are installed in the PC, one reset connection is sufficient as shown on the figure. Two PC reset connectors are provided on the frame grabbers to help cabling. The picture below shows a typical system.



## 3.2. Software application

### MultiCam API

The MultiCam API is a unified application programming interface provided with Euresys frame grabbers. Access to the frame grabber parameters is performed through software objects defined in MultiCam.

One object is dedicated to control common features of all installed frame grabbers. This “system wide” object is called **configuration**. Refer to the MultiCam documentation for more details about the configuration object.

When setting a watchdog parameter in the configuration object, the parameter is valid for all boards in the system. The typical procedure described below is therefore valid for any system, independently of the number of installed frame grabbers.

### Typical procedure

A typical procedure is described below:

#### When the application starts:

- 1) Initialize the driver (see `McOpenDriver` in the MultiCam documentation).
- 2) Start the Watchdog Application Monitor:

```
// A timeout of 5 seconds is chosen for the application
McSetParamInt (MC_CONFIGURATION, MC_ApplicationTimeOut_s,5);
```

- 3) Advise the Watchdog Start-up Monitor that the application has been correctly launched (the start-up procedure has been correctly completed).

```
// A timeout of 10 minutes is adequate for the next start-up
McSetParamInt (MC_CONFIGURATION, MC_StartupTimeout_s, 600);
```

#### During application running:

- 1) Advise the Watchdog Application Monitor that the application is still running.

```
// The code line below should be at least executed every 5
// seconds to avoid closure of the PC reset relay.
// Note: the MultiCam get function can be used to retrieve
// the ApplicationTimeout_s configured when the application
// has been started.
McSetParamInt (MC_CONFIGURATION, MC_ApplicationTimeout_s,5);
```

### When the application ends:

- 1) Disable the Watchdog Application Monitor.

```
// A dedicated constant (MC_DISABLE) is used to disable the
// watchdog. In that case, the corresponding monitor goes in
// the OFF state
McSetParamInt (MC_CONFIGURATION, MC_ApplicationTimeout_s,
               MC_DISABLE);
```

- 2) Terminate the driver (see `McCloseDriver` in the MultiCam documentation).

## 3.3. Hints

Latter frame grabbers provide an advanced I/O subsystem implemented with a microcontroller. The advanced I/O subsystem manages the watchdog and the input/output. Some of the additional useful features described below are only available with the advanced I/O subsystem.

### Watchdog resets logging

The advanced I/O subsystem counts the occurrences of watchdog resets and delivers the result in the `WatchdogResetsCount` parameter. This is a way to count the number of malfunctions the system has encountered.

The counter can be reset by setting it to 0 with the `MCSetParamInt` (or `MCSetParamNmInt`) function.

### Partial watchdog usage

When the Start-up Monitor is not required, there is no need to explicitly disable it, just ignore the `StartupTimeout_s` parameter.

Similarly, when the Application Monitor is not required, there is no need to explicitly disable it, just ignore the `ApplicationTimeout_s` parameter.

However, full protection requires the use of both parameters.

### Remote signaling

In a highly secured application, a remote monitoring of the correct operation of the system is required. The boards equipped with the advanced I/O subsystem allow this easily.

The programmer has to set a solid-state relay output to LOW (CLOSED) at the beginning of the application (See the "Euresys Hardware Documentation" for the exact procedure). A closed state means that the application is running.

If a watchdog reset occurs, the watchdog closes the PC reset relay **and** opens all solid-state relays, therefore advising a remote supervisor that a watchdog reset has occurred.

### Dealing with exceptional long boot

For several reasons, the operating system can decide to perform exceptional task such as check-disk during start-up. This leads to exceptional long boot not always foreseeable.

During a long boot, the Start-up monitor resets the PC since the timeout period expires.

The key aspect of the Start-up Monitor is that it goes in the OFF state after generating a watchdog reset and hence generates only one reset during a long boot. The second long boot will have enough time to complete successfully.

### Minimum watchdog timeout

Frame grabbers provided with the advanced I/O subsystem offer the possibility to configure timeouts from one second to more than four hours.

Frame grabbers without the advanced I/O subsystem do not support timeouts below five minutes.

Please read the “Euresys Hardware Documentation” for information about the frame grabber used.

## 3.4. Watchdog control parameters summary

The table below summarizes the parameters related to the watchdog control.

<i>Watchdog Control Parameters</i>	<i>Meaning</i>
StartupTimeout_s	Defines the maximum time (in second) that the watchdog allows the system to start up and launch the application. Setting this parameter also advises the Start-up Monitor that the application has correctly been launched after start-up.
ApplicationTimeout_s	Defines the maximum time (in second) that the watchdog allows the application to periodically signal its correct operation. Setting this parameter also advises the Application Monitor that the application is running properly.
WatchdogResetsCount	The parameter is incremented by 1 each time a watchdog reset is generated. The parameter can be cleared by setting it to 0.

## 4. Glossary

### Application

The application is the program running on the PC and responsible for acquiring, storing and/or transferring captured images.

### Application Monitor

The Application Monitor is a part of the watchdog responsible for monitoring the system when the application is running.

### Euresys Hardware Documentation

The Euresys Hardware Documentation is a help file typically located in *Start → All Programs → Euresys MultiCam → MultiCam Documentation*.

### MultiCam

MultiCam is a software tool designed to assist the programmer in controlling Euresys frame grabbers.

### PC

The PC designates the personal computer used to build the system.

### PC Reset Relay

The PC reset relay is a component of the watchdog installed on Euresys frame grabber . It is closed when the PC has to be reset.

### Programmer

The programmer is the writer of the application.

### Software Application

See Application.

### Start-up

The start-up is a sequence of operations starting from the booting of the PC and ending with the launch of the application.

### Start-up Monitor

The Start-up Monitor is a part of the watchdog responsible for monitoring the system from the booting of the PC to the launching of the application.

### System

The system is made up (in this context) of the PC, the frame grabbers and the application running on the PC.

### Watchdog

The watchdog is an independent hardware device implemented on a frame grabber responsible for monitoring the correct operation of the system and able to reset the PC in case of malfunction.

**Copyright Euresys s.a.**  
**2005**