HighIntegritySystems

WITTENSTEIN

Watchdog Strategies for RTOS enabled embedded systems

A Watchdog timer is an **electronic timer** that is **used to detect and recover from errors within embedded systems**. The basic principle of the Watchdog timer is simple but effective. Within a specific time-period, the system has to notify the Watchdog that it is still operational. If the Watchdog does not receive this notification then it assumes there has been a failure and places the system into a known state. Typically, the Watchdog will reset the processor. However, for systems that are more complex the Watchdog may have to trigger a series of operations to place the system into a known safe state. Many processors support on-chip Watchdog functionality, but, for extra security, some designers prefer to use a separate discrete Watchdog component.

The challenge faced by embedded software developers is deciding when to notify the Watchdog that the system is still functional. This is more complicated when using a pre-emptive RTOS, as the software is broken down into individual Tasks operating independently. Now the designer needs to consider carefully what constitutes a working system.

Basic Watchdog Protection

In a basic system, the designer may choose to have a periodic Task that simply notifies the Watchdog at the required frequency. In this scenario, the system remains available providing the Task that refreshes the Watchdog operates at the correct frequency. A complete systems crash or a failure within the Task notifying the Watchdog would cause the Watchdog to expire, placing the system into a safe state. However, if the system fails in a way whereby the Task that notifies the Watchdog remains operational but other mission critical Tasks fail to operate, the Watchdog would not place the system into a safe state.

Improving Robustness

An improvement to this simple system would be to notify the Watchdog that the system is OK only if all Tasks have been active during the last Watchdog refresh time-period. In this case Tasks would register with a Monitor Task so that each time a Task runs, it informs the Monitor Task. When triggered, the Monitor Task will check that all registered Tasks have operated during the last time-period; if they have, the Monitor Task will notify the Watchdog that the system remains operational.

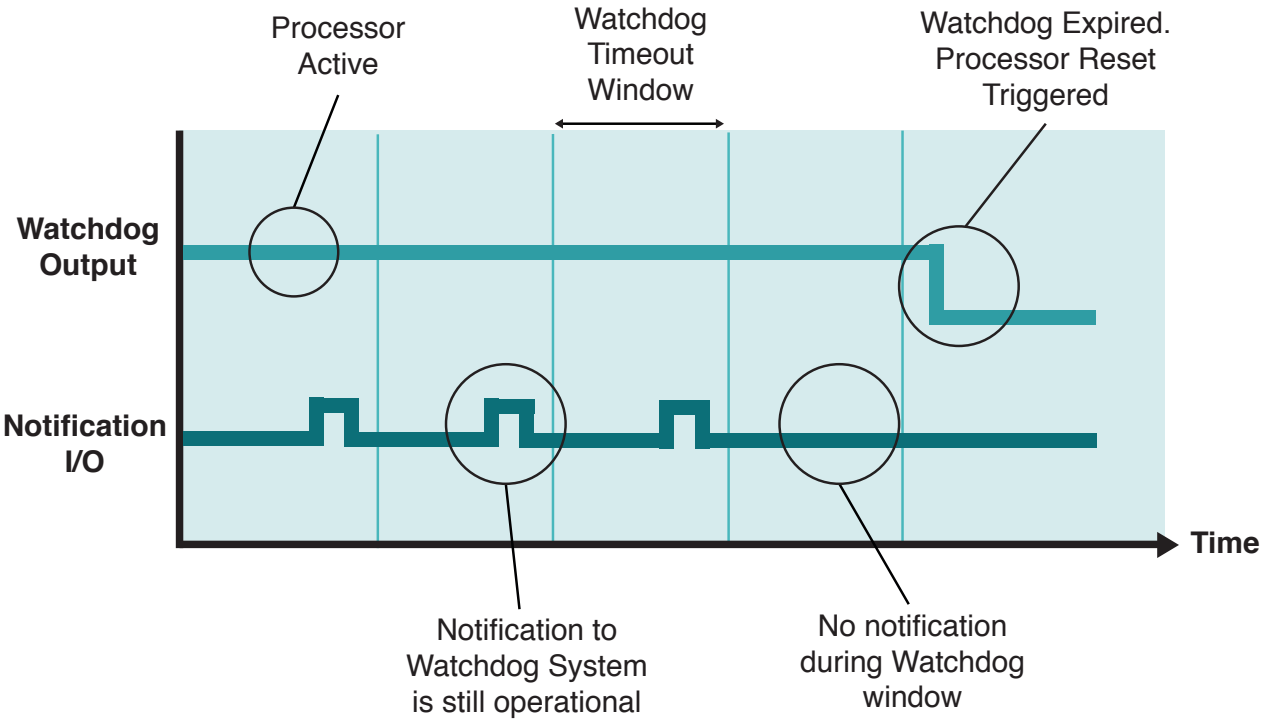

Figure 1. Watchdog Timing Diagram

WITTENSTEIN high integrity systems
Americas: +1 408 625 4712
ROTW: +44 1275 395 600

email: sales@high**integrity**systems.com
web: www.high**integrity**systems.com

To manage Tasks operating at a higher or lower frequency than the Watchdog refresh rate, the designer would need to include a time profile for all Tasks. For each Watchdog refresh period, the Monitor Task would confirm that only the expected scheduled Tasks have been active. In some systems however, knowing the Task is still operational does not provide sufficient assurance that the system is still operating correctly. In these systems the operation of critical code sections must be monitored as well as the Tasks they are found within.

To complete the temporal monitoring of events, an additional enhancement would be to monitor the response time of Interrupt Service Routines (ISR). Here the Monitor Task would measure the actual time it takes to process the response, from the time the ISR is triggered to the time when the overall operation has completed.

By monitoring the timing profile of individual Tasks, critical code sections and ISR response times, the designer has a high level of assurance that the Watchdog notification mechanism is working as expected. However, the complexity of the Monitor Task has increased significantly.

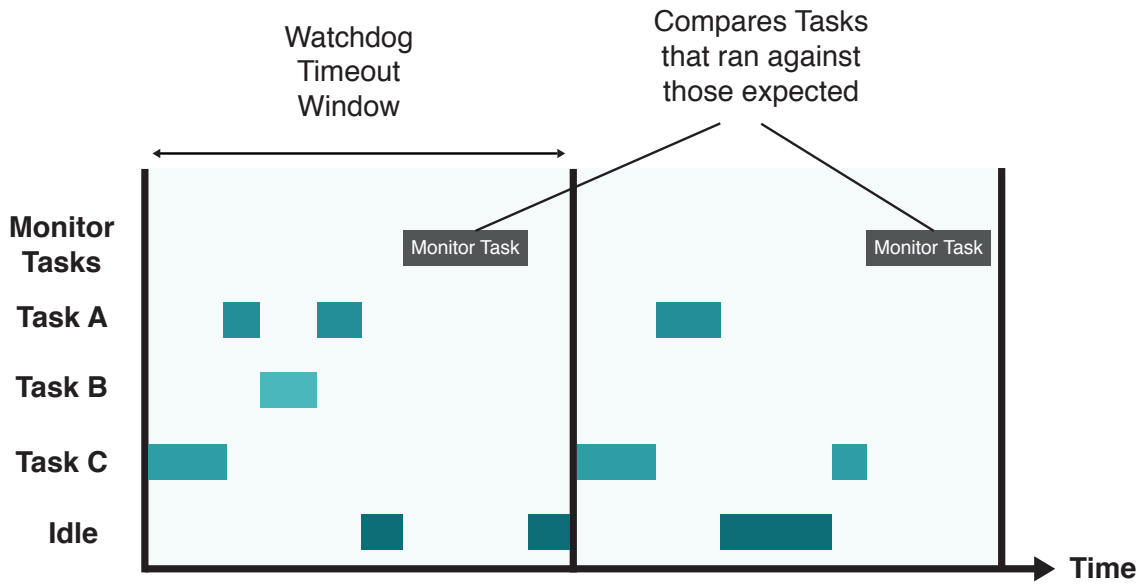

Figure 2. Monitor Task Scheduling

Sophisticated Task Monitoring

SAFE**Checkpoints** from WITTENSTEIN high integrity systems is a software component that provides this sophisticated Task monitoring capability, ensuring the scheduling of tasks is occurring as intended.

The Checkpoints mechanism allows the user to specify timing tolerances for critical sections of code. This can be used to ensure that:

- Periodic Tasks run within tolerances.
- Sections of processing within Tasks complete.
- ISR execution to Handler Task processing completes with allowable tolerances.
- Complex functionality involving multiple tasks completes within allowable tolerances.

Individual Checkpoints can specify their own callback function or the system error hook can be activated.

- Single shot and Periodic checkpoints can be created.
- Periodic checkpoints can operate in fixed or relative timing modes.

SAFE**Checkpoints** is available with a full Design Assurance Pack supporting certification to IEC 61508 SIL 3, and is delivered fully integrated with either SAFE**RTOS**® or SAFE**RTOS** CORE.

WITTENSTEIN high integrity systems
Americas: +1 408 625 4712
ROTW: +44 1275 395 600

email: sales@high**integrity**systems.com
web: www.high**integrity**systems.com