SAFE**RTOS**® for HY-TTC 500 Family

High Performance Off-Highway Electronic Control Units

Key Benefits

- High level of robustness and responsiveness for customer applications
- Efficient CPU usage by splitting *"main loop"* application into multiple working tasks
- Fast response to safety events due to SAFE**RTOS**® runtime performance
- Reduced costs due to lower development and maintenance efforts

Derived from the widely known FreeRTOS functional model, SAFE**RTOS**® was designed for safety-critical use and allows to implement the application as a set of independent working tasks. The HY-TTC 500 SAFE**RTOS**® integration extends the SAFE**RTOS**® functionality with control application specific features, including runtime separation into safety-critical and non-safety-critical tasks and a monitoring concept for ensuring timely execution of all tasks.
The HY-TTC 500 controllers including SAFE**RTOS**® integration are safety certified to fulfill safety requirements up to SIL 2 (IEC 61508) and PL d (ISO 13849).

Features

- Designing applications as a set of independent tasks, each with an assigned priority
- Increased flexibility in application implementation leading to smaller code that is more modular, easier to test and to maintain.
- Safety-critical and non-safety-critical task isolation and separation
- Constant monitoring of each task's execution time and responsiveness
- Application task ownership of I/O-Pins
- Efficient inter-task communication and synchronization mechanisms

Application Fields

- Large construction / material handling machines
- Large municipal vehicles
- Large agricultural machines

www.ttcontrol.com

SAFE**RTOS**®

SAFE**RTOS**® is a safety certified Real Time Operating System (RTOS) for embedded processors. It delivers superior performance and pre-certified dependability, whilst utilizing minimal resources.

An RTOS application is designed as a set of independent tasks, each task having an assigned priority. A part of the operating system, called the scheduler, is responsible for deciding which task to execute at what time. It provides the impression and effect of simultaneous execution by rapidly switching between tasks.

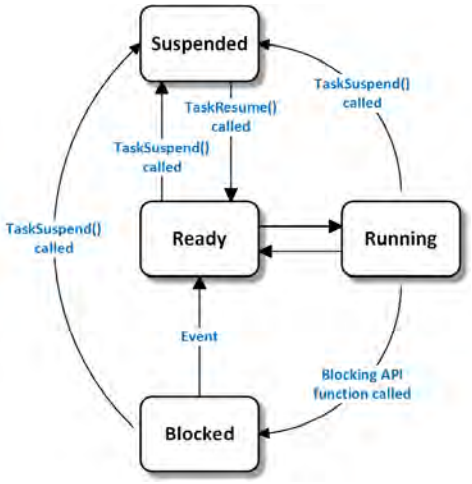

Figure: States of a task in an RTOS

The SAFE**RTOS**® scheduling policy is fixed-priority preemptive (the scheduler ensures that the processor executes the highest priority task of all the tasks that are ready to execute).

The "*main loop*" application is split into multiple working tasks that run until they are interrupted by a higher priority task, blocked (waiting for an external event) or until a time limit expired. While waiting for the external event or their next turn to run, other tasks are allowed to execute, which enables faster control loops leading to an increased efficiency of CPU usage.

Safety-Critical and Non-Safety-Critical Task Separation

The task isolation and separation functionality added by TTControl, using the processor's Memory Protection Unit (MPU), prevents the risk of unwanted interleave of application tasks with tasks performing other types of functionality (e.g. low-level accesses to peripheral control).

Up to 29 user tasks can be created, classified either as safety-critical or non-safety-critical task.

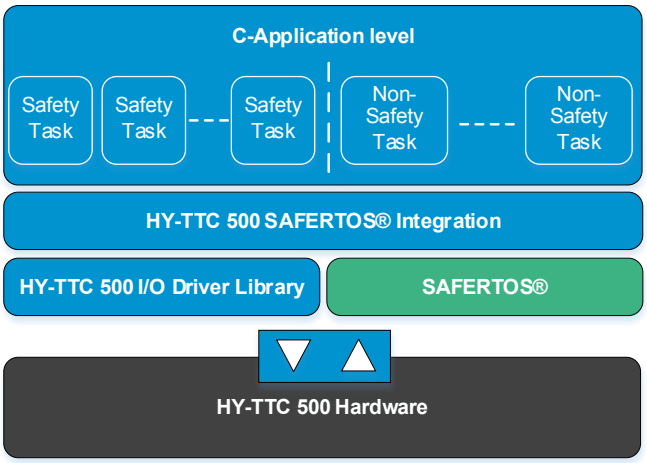

Figure: HY-TTC 500 SAFE**RTOS**® integration diagram

I/O Access and Ownership

HY-TTC 500 I/O driver library integration provides full HY-TTC 500 family I/O capability, similar to the CODESYS and C-programming environments.

Task access to the I/O-Pins is controlled by the ownership concept. All application tasks are allowed to read data from an input I/O-Pin regardless of ownership, but only the owner can change the configuration of an I/O-Pin. In case of an output pin, only the task owning the I/O-Pin is allowed to set data.

To fulfill safety requirements, the input and outputs with safety configuration cannot be owned by non-safety-critical tasks. An implemented control mechanism ensures the consistency of output data.

Task Communication and Monitoring

The efficient inter-task communication and synchronization mechanism using queues permits data to be safely transferred between tasks.

All tasks (both safety- and non-safety-critical) are monitored from their time execution perspective.

Out-of-time responses are reported to the application level using call-back functions. The decision regarding actions to be taken is up to the user application (i.e. ignore, enter safe state or stop a non-safety-critical software task).

TTControl Italy, Brixen
Phone: +39 0472 26 80-11

TTControl Austria, Vienna
Phone: +43 1 585 34 34-0