

Sommario

SY10EnterCriticalSection 3
IMPLEMENTATION 3

SY10EnterCriticalSection

SY = System function

The SY10EnterCriticalSection function puts the calling task unit in a critical section.

IMPLEMENTATION

SY10EnterCriticalSection(lockPool)

Parameters:

| IN/OUT | VARIABLE TYPE | EXAMPLE NAME | DIM | |
|--------|---------------|--------------|-----|--|
| IN | CONST | unitID | L | Caller ID unit task |
| IN | ARRGBL | pool | B | Critical section pools informations |
| IN/OUT | GLOBAL | locked | F | Lock state (0 = unlocked, 1 = locked) |
| IN | CONST | blocking | L | Lock mode (0 = non-blocking, 1 = blocking) |

Description:

Through the SY10EnterCriticalSection call the unit task caller asks the system to enter a critical section.

If the critical section is free caller will be assigned to the task by returning the State locked to 1. If the task that requires you to enter the critical section is already entered, the function will return the State locked to 1 incrementing the counter of requests for the task in progress.

If the critical section is already occupied by another unit task and the blocking mode is set to 0 and you will have immediate exit with status 0 locked.

Otherwise based on the blocking you will have the return with locked state 0 or the unit task is put on hold and run automatically one context switch.

Example

The following example initializes an information pool used then to a critical section.

task unit:

```

;===
; Unit B
;
GLOBAL
  Counter          L   INOUT
; local variables
CONST
  UNIT_ID          2
GLOBAL
  locked           F   L
  looping          L
; main entry point
BEGIN
  CALL TASK_INIT
  WHILE TRUE
    CALL TASK_EXECUTE
    WAIT A_LOOP
  ENDWHILE
END

;===
; Task initialization
;
SUB TASK_INIT
  WAIT INIT.Initialized
ENDSUB

;===
; Task execution
;
SUB TASK_EXECUTE
  SY10EnterCriticalSection(UNIT_ID, LockPool, locked, TRUE)
  CALL COUNTER_ADD
  SY10LeaveCriticalSection(UNIT_ID, LockPool, locked, TRUE)
ENDSUB

;===
; Counter add
;
SUB COUNTER_ADD
  Counter = Counter + 1
  ; this simulate an automatic context switch for a device write and read access
  WAIT A_LOOP
ENDSUB

```

Documento generato automaticamente da **Qem Wiki** - <https://wiki.qem.it/>
Il contenuto wiki è costantemente aggiornato dal team di sviluppo, è quindi possibile che la versione online contenga informazioni più recenti di questo documento.