µC/OS-II, The Real-Time Kernel

V2.92.11 Quick Reference Chart

Legend:
Black
Orange
Red
Blue
Green

Micriμm
1290 Weston Road, Suite 306
Weston, FL 33326
USA
www.Micrium.com

Semaphores (OS_SEM.C)

```
INT16U      OSSemAccept     (OS_EVENT     *pevent);          Commentaires  G.B.
OS_EVENT    *OSSemCreate    (INT16U        cnt);             Non Bloquant
OS_EVENT    *OSSemDel       (OS_EVENT     *pevent,           Il faut initialisé le sémaphore
                             INT8U          opt,
                             INT8U         *perr);

void        OSSemPend       (OS_EVENT     *pevent,           Bloquant
                             INT32U         timeout,
                             INT8U         *err);

INT8U       OSSemPendAbort  (OS_EVENT     *pevent,           Permet de libérer la tâche la plus
                             INT8U          opt,             prioritaire en attente ou toutes
                             INT8U         *perr);           les tâches (voir les 2 options)

INT8U       OSSemPost       (OS_EVENT     *pevent);

INT8U       OSSemQuery      (OS_EVENT     *pevent,           OS_SEM_DATA:
                             OS_SEM_DATA   *p_sem_data);

                                                            N.B.
                                                            - la sémaphore binaire n'est une vrai
                                                            sémaphore binaire (pas de plafond à 1,
                                                            il fait utiliser OSSemAccept)
                                                            - peut aussi servir de sémaphore compteur
                                                            - si on utilise pour exclusion mutuelle,
                                                            il n'y aura pas de mécanismes d'héritage
                                                            de priorité

void        OSSemSet        (OS_EVENT     *pevent,           Permet de ré-initialiser à tout
                             INT16U         cnt,             moment de l'exécution
                             INT8U         *perr);
```

OPTIONS (opt)

Miscellaneous

µC/OS-II, The Real-Time Kernel

V2.92.11 Quick Reference Chart

Legend:
Black
Orange
Red
Blue
Green

Micriμm
1290 Weston Road, Suite 306
Weston, FL 33326
USA
www.Micrium.com

Mutual Exclusion Semaphores (OS_MUTEX.C)

```
BOOLEAN        OSMutexAccept     (OS_EVENT      *pevent,
                                  INT8U         *perr);
OS_EVENT      *OSMutexCreate     (INT8U          prio,
                                  INT8U         *perr);
OS_EVENT      *OSMutexDel        (OS_EVENT      *pevent,
                                  INT8U          opt,
                                  INT8U         *perr);
void           OSMutexPend       (OS_EVENT      *pevent,
                                  INT32U         timeout,
                                  INT8U         *perr);
INT8U          OSMutexPost       (OS_EVENT      *pevent);
INT8U          OSMutexQuery      (OS_EVENT      *pevent,
                                  OS_MUTEX_DATA *p_mutex_data);
```

Commentaires G.B.
Non Bloquant

prio sert à déterminer la prorité qui sera donnée en héritage. Cette priorité doit être plus grane (inférieur ou sens uC) à toutes les tâches qui compétitionnent pour ce mutex.

Bloquant

N.B.
- Contrairement au sémaphore la valeur du mutex est toujours à 1 en partant et ne peut être ré-initialisée à 1

OPTIONS (opt)

Miscellaneous

OS_MUTEX_DATA:

µC/OS-II, The Real-Time Kernel

V2.92.11 Quick Reference Chart

Legend:
Black
Orange
Red
Blue
Green

Micriμm
1290 Weston Road, Suite 306
Weston, FL 33326
USA
www.Micrium.com

Event Flags (OS_FLAG.C)

					OPTIONS (opt)	Miscellaneous
				Commentaires G.B.		
OS_FLAGS	OSFlagAccept	(OS_FLAG_GRP	*pgrp,	Non Bloquant		
		OS_FLAGS	flags,			
		INT8U	wait_type,			
		INT8U	*perr);			
OS_FLAG_GRP	*OSFlagCreate	(OS_FLAGS	flags,			
		INT8U	*perr);			
OS_FLAG_GRP	*OSFlagDel	(OS_FLAG_GRP	*pgrp,			
		INT8U	opt,			
		INT8U	*perr);			
INT8U	OSFlagNameGet	(OS_FLAG_GRP	*pgrp,	Si on veut connaitre le nom (chaine de caractères)		
		INT8U	**pname,			
		INT8U	*perr);			
void	OSFlagNameSet	(OS_FLAG_GRP	*pgrp,	Si on veut donner un nom (chaine de caractères)		
		INT8U	*pname,			
		INT8U	*perr);			
OS_FLAGS	OSFlagPend	(OS_FLAG_GRP	*pgrp,	Bloquant		
		OS_FLAGS	flags,			
		INT8U	wait_type,			
		INT16U	timeout,			
		INT8U	*perr);	N.B. - OS_FLAG_WAIT_SET_ALL + OS_FLAG_CONSUME est équivalent à attendre avec plusieurs sémaphores mais moins d'espace et permet l'utilisation de masques - Si consume ne fonctionne pas faire de OSFlagPost avec OS_FLAG_CLR		
OS_FLAGS	OSFlagPendGetFlagsRdy	(void);				
OS_FLAGS	OSFlagPost	(OS_FLAG_GRP	*pgrp,	Permet de connaitre le flag qui a permis le dernier déblocage		
		OS_FLAGS	flags,			
		INT8U	opt,			
		INT8U	*perr);			
OS_FLAGS	OSFlagQuery	(OS_FLAG_GRP	*pgrp,			
		INT8U	*perr);			

µC/OS-II, The Real-Time Kernel

V2.92.11 Quick Reference Chart

Legend:
Black
Orange
Red
Blue
Green

Micriμm
1290 Weston Road, Suite 306
Weston, FL 33326
USA
www.Micrium.com

Message Mailboxes (OS_MBOX.C)

				Commentaires G.B.
void	*OSMboxAccept	(OS_EVENT	*pevent);	Non Bloquant
OS_EVENT	*OSMboxCreate	(void	*msg);	
OS_EVENT	*OSMboxDel	(OS_EVENT INT8U INT8U	*pevent, opt, *perr);	
void	*OSMboxPend	(OS_EVENT INT32U INT8U	*pevent, timeout, *perr);	Bloquant
INT8U	OSMboxPendAbort	(OS_EVENT INT8U INT8U	*pevent, opt, *perr);	Permet de libérer la tâche la plus prioritaire en attente ou toutes les tâches (voir les 2 options)
INT8U	OSMboxPost	(OS_EVENT void	*pevent, *pmsg);	Non Bloquiant même si le même si le MBox est plein. Il faut vérifier avec le message de retour (OS_MBOX_FULL)
INT8U	OSMboxPostOpt	(OS_EVENT void INT8U	*pevent, *pmsg, opt);	Comme OSMBoxPost mais permet de faire 1) un broadcast à toutes les tâches en attente ou 2) demandé qu'il n'y est pas d'appel au scheduler après la production du message. Si OS_POST_OPT_NONE, ça revient à OSMBoxPost.
INT8U	OSMboxQuery	(OS_EVENT OS_MBOX_DATA	*pevent, *p_mbox_data);	N.B. Le mailbox est un cas particulier de la queue avec un seul élément. On a donc pas à maintenir la structure de données de la queue

OPTIONS (opt)

Miscellaneous

OS_MBOX_DATA:

µC/OS-II, The Real-Time Kernel

V2.92.11 Quick Reference Chart

Legend:
Black
Orange
Red
Blue
Green

Micriμm
1290 Weston Road, Suite 306
Weston, FL 33326
USA
www.Micrium.com

Message Queues (OS_Q.C)

```
void        *OSQAccept     (OS_EVENT  *pevent,
                            INT8U      *perr);
OS_EVENT    *OSQCreate     (void       **start,
                            INT16U      size);
OS_EVENT    *OSQDel        (OS_EVENT  *pevent,
                            INT8U       opt,
                            INT8U      *perr);
INT8U        OSQFlush      (OS_EVENT  *pevent);
void        *OSQPend       (OS_EVENT  *pevent,
                            INT32U      timeout,
                            INT8U      *perr);
INT8U        OSQPendAbort  (OS_EVENT  *pevent,
                            INT8U       opt,
                            INT8U      *perr);
INT8U        OSQPost       (OS_EVENT  *pevent,
                            void       *pmsg);
INT8U        OSQPostFront  (OS_EVENT  *pevent,
                            void       *pmsg);
INT8U        OSQPostOpt    (OS_EVENT  *pevent,
                            void       *pmsg,
                            INT8U       opt);
INT8U        OSQQuery      (OS_EVENT  *pevent,
                            OS_Q_DATA  *p_q_data);
```

Commentaires G.B.

Non Bloquant

Bloquant

Permet de libérer la tâche la plus prioritaire en attente ou toutes les tâches (voir les 2 options)

Non Bloquiant même si le même si le Q est plein. Il faut vérifier avec le message de retour (OS_MBOX_FULL)

Comme OSMQPost mais permet: 1) de faire un broadcast à toutes les tâches en attente ou 2) demandé qu'il n'y est pas d'appel scheduler après la production du message ou 3) écrire en début de Q (FIFO). Si OS_POST_OPT_NONE, ça revient à OSMBoxPost.

N.B. Q est une queue de message circulaire (LIFO) mais peut aussi être utilisé en FIFO.

OPTIONS (opt)

Miscellaneous

OS_Q_DATA:

µC/OS-II, The Real-Time Kernel

V2.92.11 Quick Reference Chart

Legend:
Black
Orange
Red
Blue
Green

Micriμm
1290 Weston Road, Suite 306
Weston, FL 33326
USA
www.Micrium.com

Memory Management (OS_MEM.C)

```
OS_MEM      *OSMemCreate        (void          *addr,
                                 INT32U         nblks,
                                 INT32U         blksize,
                                 INT8U          *perr);
void        *OSMemGet           (OS_MEM        *pmem,
                                 INT8U          *perr);
INT8U        OSMemNameGet       (OS_MEM        *pmem,
                                 INT8U         **pname,
                                 INT8U          *perr);
void         OSMemNameSet       (OS_MEM        *pmem,
                                 INT8U          *pname,
                                 INT8U          *perr);
INT8U        OSMemPut           (OS_MEM        *pmem,
                                 void           *pblk);
INT8U        OSMemQuery         (OS_MEM        *pmem,
                                 OS_MEM_DATA    *p_mem_data);
```

OPTIONS (opt)

Miscellaneous

```
OS_MEM_DATA:
```

µC/OS-II, The Real-Time Kernel

V2.92.11 Quick Reference Chart

Legend:
Black
Orange
Red
Blue
Green

Micriμm
1290 Weston Road, Suite 306
Weston, FL 33326
USA
www.Micrium.com

Task Management (OS_TASK.C)

```
INT8U          OSTaskChangePrio      (INT8U          oldprio,
                                      INT8U          newprio);
INT8U          OSTaskCreate          (void          (*task)(void *p_arg),
                                      void          *p_arg,
                                      OS_STK        *ptos,
                                      INT8U          prio);
INT8U          OSTaskCreateExt       (void          (*task)(void *p_arg),
                                      void          *p_arg,
                                      OS_STK        *ptos,
                                      INT8U          prio,
                                      INT16U         id,
                                      OS_STK        *pbos,
                                      INT32U         stk_size,
                                      void          *pext,
                                      INT16U         opt);


INT8U          OSTaskDel             (INT8U          prio);
INT8U          OSTaskDelReq          (INT8U          prio);
INT8U          OSTaskNameGet         (INT8U          prio,
                                      INT8U         **pname,
                                      INT8U          *perr);
void           OSTaskNameSet         (INT8U          prio,
                                      INT8U          *pname,
                                      INT8U          *perr);
INT32U         OSTaskRegGet          (INT8U          prio,
                                      INT8U          id,
                                      INT8U          *perr);
INT8U          OSTaskRegGetID        (INT8U          *perr);
void           OSTaskRegSet          (INT8U          prio,
                                      INT8U          id,
                                      INT32U         value,
                                      INT8U          *perr);
INT8U          OSTaskResume          (INT8U          prio);
INT8U          OSTaskSuspend         (INT8U          prio);
INT8U          OSTaskStkChk          (INT8U          prio,
                                      OS_STK_DATA    *p_stk_data);


INT8U          OSTaskQuery           (INT8U          prio,
                                      OS_TCB         *p_task_data);
```

OPTIONS (opt)

Miscellaneous

```
OS_STK_DATA:
```

µC/OS-II, The Real-Time Kernel

Legend:
Black
Orange
Red
Blue
Green

Micriμm
1290 Weston Road, Suite 306
Weston, FL 33326
USA
www.Micrium.com

OPTIONS (opt) **Miscellaneous**

µC/OS-II, The Real-Time Kernel

V2.92.11 Quick Reference Chart

Legend:
Black
Orange
Red
Blue
Green

Micriμm
1290 Weston Road, Suite 306
Weston, FL 33326
USA
www.Micrium.com

Timer Management (OS_TMR.C)

```
OS_TMR        *OSTmrCreate          (INT32U           dly,
                                     INT32U           period,
                                     INT8U            opt,
                                     OS_TMR_CALLBACK  callback,
                                     void            *callback_arg,
                                     INT8U           *pname,
                                     INT8U           *perr);
BOOLEAN        OSTmrDel             (OS_TMR          *ptmr,
                                     INT8U           *perr);
INT8U          OSTmrNameGet         (OS_TMR          *ptmr,
                                     INT8U          **pdest,
                                     INT8U           *perr);
INT32U         OSTmrRemainGet       (OS_TMR          *ptmr,
                                     INT8U           *perr);
INT8U          OSTmrStateGet        (OS_TMR          *ptmr,
                                     INT8U           *perr);
BOOLEAN        OSTmrStart           (OS_TMR          *ptmr,
                                     INT8U           *perr);
void           OSTmrStop            (OS_TMR          *ptmr,
                                     INT8U            opt,
                                     void            *callback_arg,
                                     INT8U           *perr);
void           OSTmrSignal          (void);
```

OPTIONS (opt)

Miscellaneous

µC/OS-II,

Legend:
Black
Orange
Red
Blue
Green

Micriμm
1290 Weston Road, Suite 306
Weston, FL 33326
USA
www.Micrium.com

OPTIONS (opt) **Miscellaneous**

µC/OS-II, The Real-Time Kernel

V2.92.11 Quick Reference Chart

Legend:
Black
Orange
Red
Blue
Green

Micriμm
1290 Weston Road, Suite 306
Weston, FL 33326
USA
www.Micrium.com

	OPTIONS (opt)	Miscellaneous

Port Functions (OS_CPU_A.ASM)

```
void          OSCtxSw              (void);
void          OSIntCtxSw           (void);
void          OSStartHighRdy       (void);
```

Port Functions (OS_CPU_C.C)

```
void          OSInitHookBegin      (void);
void          OSInitHookEnd        (void);
void          OSTaskCreateHook     (OS_TCB        *ptcb);
void          OSTaskDelHook        (OS_TCB        *ptcb);
void          OSTaskIdleHook       (void);
void          OSTaskReturnHook     (OS_TCB        *ptcb);
void          OSTaskStatHook       (void);
OS_STK       *OSTaskStkInit        (void          (*task)(void *p_arg),
                                    void          *p_arg,
                                    OS_STK        *ptos,
                                    INT16U         opt);
void          OSTaskSwHook         (void);
void          OSTCBInitHook        (OS_TCB        *ptcb);
void          OSTimeTickHook       (void);
```