µC/OS-II, The Real-Time Kernel

V2.85 Quick Reference Chart

Legend:
Black is for seldom used functions
Orange is for CREATE functions
Red is for DELETE functions
Blue is for commonly used functions
Green is for comments

Micriμm
949 Crestview Circle
Weston, FL 33327
USA
www.Micrium.com

Semaphores (OS_SEM.C)

```
INT16U         OSSemAccept(OS_EVENT *pevent);
OS_EVENT      *OSSemCreate(INT16U cnt);
OS_EVENT      *OSSemDel(OS_EVENT *pevent, INT8U opt, INT8U *err);

void           OSSemPend(OS_EVENT *pevent, INT16U timeout, INT8U *err);
INT8U          OSSemPendAbort(OS_EVENT *pevent, INT8U opt, INT8U *err);

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

void           OSSemSet(OS_EVENT *pevent, INT16U cnt, INT8U *err);
```

OPTIONS (opt)

```
OS_DEL_NO_PEND
OS_DEL_ALWAYS

OS_PEND_OPT_NONE
OS_PEND_OPT_BROADCAST
```

Miscellaneous

```
OS_SEM_DATA:
INT16U  OSCnt
#if OS_LOWEST_PRIO <= 63
INT8U   OSEventGrp
INT8U   OSEventTbl[]
#else
INT16U  OSEventGrp
INT16U  OSEventTbl[]
#endif
```

Mutual Exclusion Semaphores (OS_MUTEX.C)

```
BOOLEAN        OSMutexAccept(OS_EVENT *pevent, INT8U *err);
OS_EVENT      *OSMutexCreate(INT8U prio, INT8U *err);
OS_EVENT      *OSMutexDel(OS_EVENT *pevent, INT8U opt, INT8U *err);

void           OSMutexPend(OS_EVENT *pevent, INT16U timeout, INT8U *err);
INT8U          OSMutexPost(OS_EVENT *pevent);
INT8U          OSMutexQuery(OS_EVENT *pevent, OS_MUTEX_DATA *p_mutex_data);
```

```
OS_DEL_NO_PEND
OS_DEL_ALWAYS
```

```
OS_MUTEX_DATA:
BOOLEAN OSValue
INT8U   OSOwnerPrio
INT8U   OSMutexPIP
#if OS_LOWEST_PRIO <= 63
INT8U   OSEventGrp
INT8U   OSEventTbl[]
#else
INT16U  OSEventGrp
INT16U  OSEventTbl[]
#endif
```

µC/OS-II, The Real-Time Kernel

V2.85 Quick Reference Chart

Legend:
Black is for seldom used functions
Orange is for CREATE functions
Red is for DELETE functions
Blue is for commonly used functions
Green is for comments

Micriμm
949 Crestview Circle
Weston, FL 33327
USA
www.Micrium.com

	OPTIONS (opt)	Miscellaneous
Event Flags (OS_FLAG.C)		
`OS_FLAGS OSFlagAccept(OS_FLAG_GRP *pgrp, OS_FLAGS flags, INT8U wait_type, INT8U *err);`		
`OS_FLAG_GRP *OSFlagCreate(OS_FLAGS flags, INT8U *err);`		
`OS_FLAG_GRP *OSFlagDel(OS_FLAG_GRP *pgrp, INT8U opt, INT8U *err);`	OS_DEL_NO_PEND OS_DEL_ALWAYS	
`INT8U OSFlagNameGet(OS_FLAG_GRP *pgrp, INT8U *pname, INT8U *err);`		
`void OSFlagNameSet(OS_FLAG_GRP *pgrp, INT8U *pname, INT8U *err);`		
`OS_FLAGS OSFlagPend(OS_FLAG_GRP *pgrp, OS_FLAGS flags, INT8U wait_type, INT16U timeout, INT8U *err);`		wait_type: OS_FLAG_WAIT_CLR_ALL OS_FLAG_WAIT_CLR_AND OS_FLAG_WAIT_CLR_ANY OS_FLAG_WAIT_CLR_OR OS_FLAG_WAIT_SET_ALL OS_FLAG_WAIT_SET_AND OS_FLAG_WAIT_SET_ANY OS_FLAG_WAIT_SET_OR + OS_FLAG_CONSUME
`OS_FLAGS OSFlagPendGetFlagsRdy(void);`		
`OS_FLAGS OSFlagPost(OS_FLAG_GRP *pgrp, OS_FLAGS flags, INT8U opt, INT8U *err);`	OS_FLAG_CLR OS_FLAG_SET	
`OS_FLAGS OSFlagQuery(OS_FLAG_GRP *pgrp, INT8U *err);`		
Message Mailboxes (OS_MBOX.C)		
`void *OSMboxAccept(OS_EVENT *pevent);`		
`OS_EVENT *OSMboxCreate(void *msg);`		
`OS_EVENT *OSMboxDel(OS_EVENT *pevent, INT8U opt, INT8U *err);`		
`void *OSMboxPend(OS_EVENT *pevent, INT16U timeout, INT8U *err);`	OS_DEL_NO_PEND OS_DEL_ALWAYS	
`INT8U OSMboxPendAbort(OS_EVENT *pevent, INT8U opt, INT8U *err);`	OS_PEND_OPT_NONE OS_PEND_OPT_BROADCAST	
`INT8U OSMboxPost(OS_EVENT *pevent, void *msg);`		
`INT8U OSMboxPostOpt(OS_EVENT *pevent, void *msg, INT8U opt);`	OS_POST_OPT_NONE OS_POST_OPT_BROADCAST OS_POST_OPT_NO_SCHED	
`INT8U OSMboxQuery(OS_EVENT *pevent, OS_MBOX_DATA *p_mbox_data);`		**OS_MBOX_DATA:** void *msg #if OS_LOWEST_PRIO <= 63 INT8U OSEventGrp INT8U OSEventTbl[] #else INT16U OSEventGrp INT16U OSEventTbl[] #endif

µC/OS-II, The Real-Time Kernel

V2.85 Quick Reference Chart

Legend:
Black is for seldom used functions
Orange is for CREATE functions
Red is for DELETE functions
Blue is for commonly used functions
Green is for comments

Micrium
949 Crestview Circle
Weston, FL 33327
USA
www.Micrium.com

Message Queues (OS_Q.C)

```
void         *OSQAccept(OS_EVENT *pevent, INT8U *err);
OS_EVENT     *OSQCreate(void **start, INT16U size);
OS_EVENT     *OSQDel(OS_EVENT *pevent, INT8U opt, INT8U *err);

INT8U         OSQFlush(OS_EVENT *pevent);
void         *OSQPend(OS_EVENT *pevent, INT16U timeout, INT8U *err);
INT8U         OSQPendAbort(OS_EVENT *pevent, INT8U opt, INT8U *err);

INT8U         OSQPost(OS_EVENT *pevent, void *msg);
INT8U         OSQPostFront(OS_EVENT *pevent, void *msg);
INT8U         OSQPostOpt(OS_EVENT *pevent, void *msg, INT8U opt);

INT8U         OSQQuery(OS_EVENT *pevent, OS_Q_DATA *p_q_data);
```

OPTIONS (opt)

```
OS_DEL_NO_PEND
OS_DEL_ALWAYS

OS_PEND_OPT_NONE
OS_PEND_OPT_BROADCAST

OS_POST_OPT_NONE
OS_POST_OPT_BROADCAST
OS_POST_OPT_FRONT
OS_POST_OPT_NO_SCHED
```

Miscellaneous

```
OS_Q_DATA:
void   *OSMsg
INT16U  OSNMsgs
INT16U  OSQSize
#if OS_LOWEST_PRIO <= 63
INT8U   OSEventGrp
INT8U   OSEventTbl[]
#else
INT16U  OSEventGrp
INT16U  OSEventTbl[]
#endif
```

Memory Management (OS_MEM.C)

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

Miscellaneous

```
OS_MEM_DATA:
void   *OSAddr
void   *OSFreeList
INT32U  OSBlkSize
INT32U  OSNBlks
INT32U  OSNFree
INT32U  OSNUsed
```

µC/OS-II, The Real-Time Kernel

V2.85 Quick Reference Chart

Legend:
Black is for seldom used functions
Orange is for CREATE functions
Red is for DELETE functions
Blue is for commonly used functions
Green is for comments

Micriμm
949 Crestview Circle
Weston, FL 33327
USA
www.Micrium.com

	OPTIONS (opt)	Miscellaneous
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);	OS_TASK_OPT_NONE OS_TASK_OPT_STK_CHK OS_TASK_OPT_STK_CLR OS_TASK_OPT_SAVE_FP	
INT8U OSTaskDel(INT8U prio);		
INT8U OSTaskDelReq(INT8U prio);		
INT8U OSTaskNameGet(INT8U prio, INT8U *pname, INT8U *err);		
void OSTaskNameSet(INT8U prio, INT8U *pname, INT8U *err);		
INT8U OSTaskResume(INT8U prio);		
INT8U OSTaskSuspend(INT8U prio);		
INT8U OSTaskStkChk(INT8U prio, OS_STK_DATA *p_stk_data);		**OS_STK_DATA:** INT32U .OSFree INT32U .OSUsed
INT8U OSTaskQuery(INT8U prio, OS_TCB *p_task_data);		
Time Management (OS_TIME.C)		
void OSTimeDly(INT16U ticks);		
INT8U OSTimeDlyHMSM(INT8U hours, INT8U minutes, INT8U seconds, INT16U milli);		
INT8U OSTimeDlyResume(INT8U prio);		
INT32U OSTimeGet(void);		
void OSTimeSet(INT32U ticks);		
void OSTimeTick(void);		

µC/OS-II, The Real-Time Kernel

V2.85 Quick Reference Chart

Legend:
Black is for seldom used functions
Orange is for CREATE functions
Red is for DELETE functions
Blue is for commonly used functions
Green is for comments

Micrium
949 Crestview Circle
Weston, FL 33327
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)

```
OS_TMR_OPT_PERIODIC
OS_TMR_OPT_ONE_SHOT
OS_TMR_OPT_NONE
OS_TMR_OPT_CALLBACK
OS_TMR_OPT_CALLBACK_ARG
```

Miscellaneous (OS_CORE.C)

```
INT8U         OSEventNameGet(OS_EVENT *pevent, INT8U *pname, INT8U *err);
void          OSEventNameSet(OS_EVENT *pevent, INT8U *pname, INT8U *err);
void          OSInit(void);
void          OSIntEnter(void);
void          OSIntExit(void);
void          OSSchedLock(void);
void          OSSchedUnlock(void);
void          OSStart(void);
void          OSStatInit(void);
INT16U        OSVersion(void);
```

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          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);
```