Appendix

November 6, 2010

Extended ROSA API

Contents

1 Dynamic tasks

1.1 Brief API list

This is a list of the dynamic task API for ROSA.

```
tcb * ROSA_taskAdd(tcb * TCB, char *id, void *taskFunction,
                   int *stack, int stackSize);
int ROSA_taskDestroy(void);
```

1.2 Detailed API list

1.2.1 ROSA_taskAdd

Prototype:	tcb * ROSA_taskAdd(tcb * TCB, char * id, void * taskFunction, int *stack, int stackSize);
Description:	Creates a TCB and install it in the ROSA kernel.
Parameters:	

- tcb * TCB - A pointer to the TCB block to be created.
- char * id - A identification for the TCB block of length NAMESIZE (default NAMESIZE = 4)
- void * taskFunction - A pointer to the function which are to be executed by the task.
- int * stack - A pointer to the task stack area.
- int stackSize - The maximum allowed stack for this task.

Return value: tcb * to the added task.

1.2.2 ROSA_taskDestroy

Prototype:	int ROSA_taskDestroy(void)
Description:	Destroy the currently executing task.
Parameters:	tcb *TCB - A pointer to the TCB to destroy.
Return value:	TRUE or FALSE depending on success.

2 Semaphores

2.1 Brief API list

This is a list of the semaphore API for ROSA.

```
void ROSA_semCreate(sem  * semaphore);
int ROSA_semDestroy(sem * semaphore);
void ROSA_semGive(sem * semaphore);
int ROSA_semTake(sem * semaphore);
```

2.2 Detailed API list

2.2.1 ROSA_semCreate

Prototype:	void ROSA_semCreate(sem * semaphore)
Description:	Creates a semaphore.
Parameters:	sem * semaphore - A pointer to the semaphore to create.
Return value:	Nothing.

2.2.2 ROSA_semDestroy

Prototype:	int ROSA_semDestroy(sem * semaphore)
Description:	Destroy a semaphore and free all associated resources. Fail if the semaphore is locked.
Parameters:	sem * semaphore - A pointer to the semaphore to destroy.
Return value:	TRUE or FALSE depending on success.

2.2.3 ROSA_semGive

Prototype:	void ROSA_semGive(sem * semaphore)
Description:	Give a semaphore back. The semaphore can only be given back by its owner.
Parameters:	sem * semaphore - A pointer to the semaphore to give back.
Return value:	Nothing.

2.2.4 ROSA_semTake

Prototype:	int ROSA_semTake(sem * semaphore)
Description:	Try to take a semaphore. Fails and returns FALSE if the semaphore is already taken by someone else.
Parameters:	sem * semaphore - A pointer to the semaphore to try to take.
Return value:	Return TRUE or FALSE depending on the success to take semaphore.

3 sysTick

3.1 Brief API list

This is a list of the sysTick API for ROSA.

```
int ROSA_sysTickGet(void);
void ROSA_sysTickWait(int ticks);
void ROSA_sysTickWaitUntil(int tick);
```

3.2 Detailed API list

3.2.1 ROSA_sysTickGet

Prototype:	int ROSA_sysTickGet(void)
Description:	Get the current system tick from the ROSA kernel.
Parameters:	None.
Return value:	The current system tick.

3.2.2 ROSA_sysTickWait

Prototype:	void ROSA_sysTickWait(int ticks)
Description:	Wait *'ticks'* number of system ticks before continuing.
Parameters:	int ticks - The number of system ticks to wait.
Return value:	Nothing.

3.2.3 ROSA_sysTickWaitUntil

Prototype:	void ROSA_sysTickWaitUntil(int tick)
Description:	Wait until the value of the system tick is equal to *'tick'* before continuing.
Parameters:	int tick - The system tick to wait for.
Return value:	Nothing.