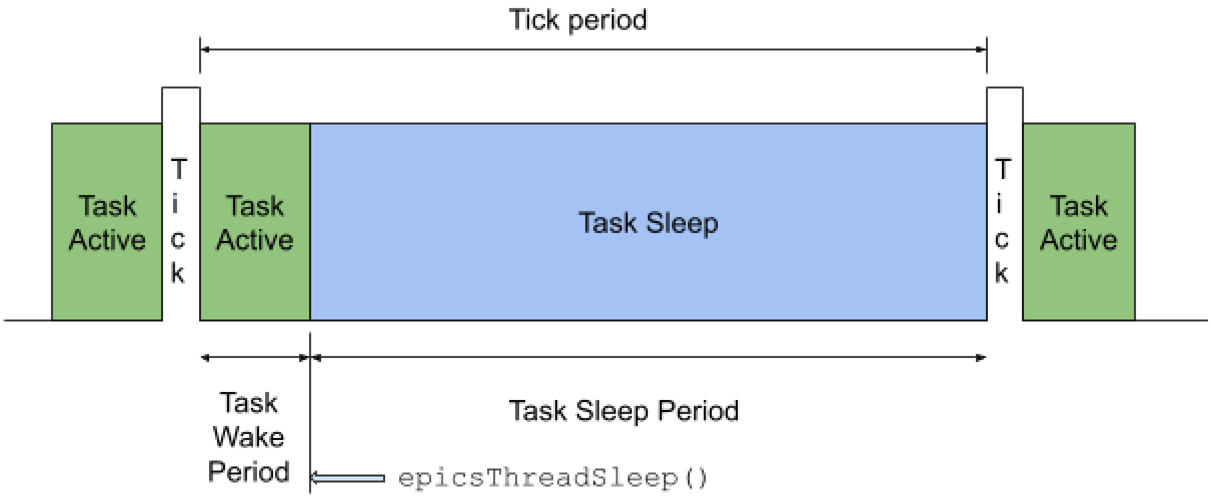
EPCIS Task Sleep On RTEMS

RTEMS uses a timer tick called from a hardware timer interrupt to manage task sleeps. This means a sleep with a time interval will align to the next while sleeping at least for the interval requested. The call is `epicsThreadSleep()` and the EPICS documentation for this call says:

> *Sleep for the specified period of time, i.e. sleep without using the cpu. If delay is >0 then the thread will sleep at least until the next clock tick. The exact time is determined by the underlying architecture. If delay is <= 0 then a delay of 0 is requested of the underlying architecture. What happens is architecture dependent but often it allows other threads of the same priority to run.*

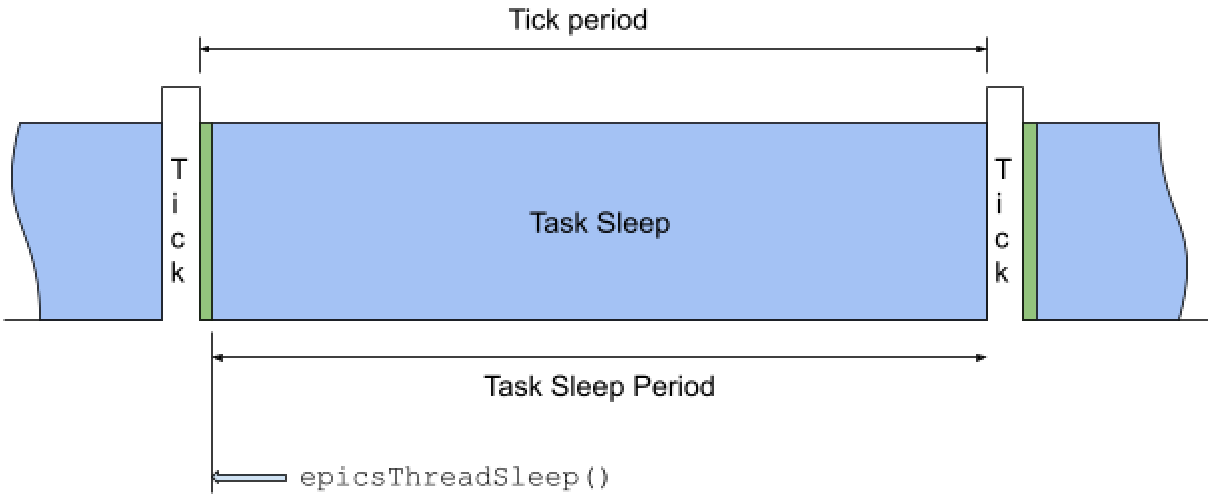
The RTEMS Classic and POSIX APIs both comply with the wording.

This picture shows the operation of a single task and a timer tick.



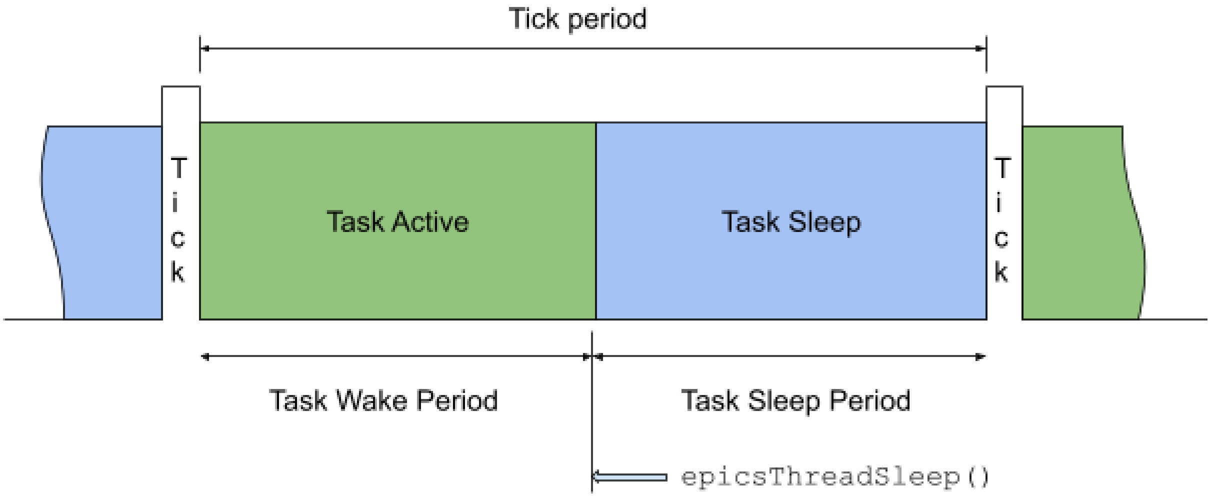
The task is active and a period of time after a tick it calls the EPICS thread sleep function. The period of time to sleep is less than the remaining time in the tick.

The EPICS test by default has the following timeline:



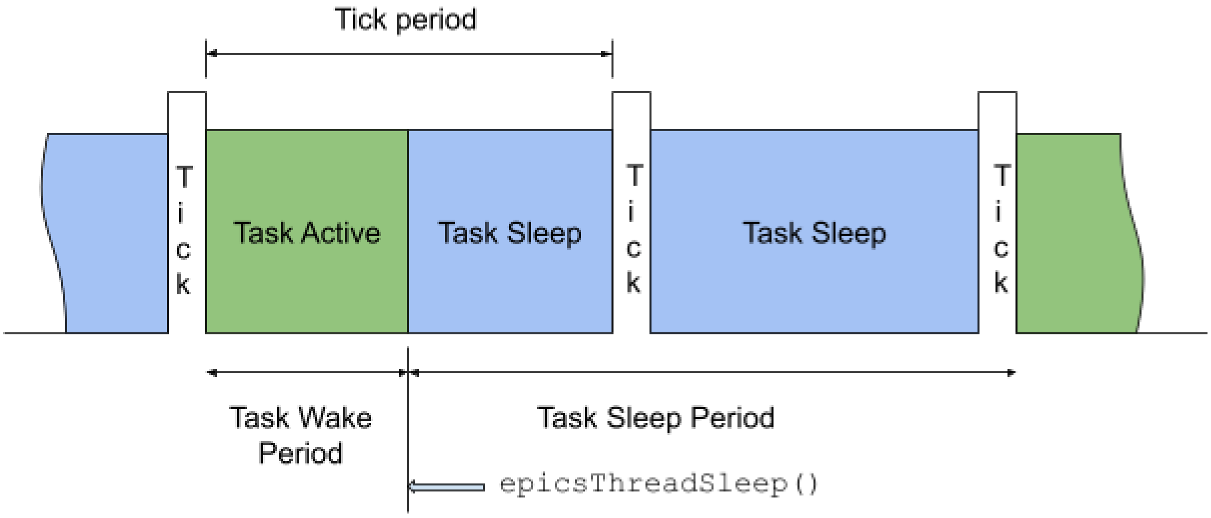
The small green task active periods are small and it is the looping logic. The classic API times off the tick so the periods and time the task sleep aligns with multiples of the tick period. When the sleep interval is the tick period the test results show very little difference between the elapsed time and the expected time.

If the task is made to be active for half the tick the functionality of the classic API and POSIX API diverge. For the classic API the timeline is:



And is valid for all intervals up to the period of the tick. The wait value can be any value greater than 0 and less than the tick period the task wakes after the next tick.

For the POSIX API this is only valid up to the period of time remaining in the tick. If the tick period is 1 millisecond and the requested sleep interval is 1 millisecond the classic API will return at the next tick and POSIX will extend the wait time to tick after the next. The POSIX's timeline is:



The remaining period of time in the tick is not enough so the period extends into the next tick.

It is important to consider these issues when reviewing the EPICS Tick Timer test and the Classic and POSIX implementations in RTEMS.