A thread in computer science is short for a thread of execution. Threads are a way for a program to divide (termed "split") itself into two or more simultaneously (or pseudo-simultaneously) running tasks.

The osThreadCreate() and osThreadDef() functions will create the threads and set their priorities respectively.

The osKernelInitialize() and osKernelStart() will setup the round robin scheduling definition for the threads and execute the kernel respectively.

osTimerThread() thread initializes and executes first.

This thread is responsible for executing time management functions specified by ARM's RTOS configuration.

The program starts executing from main(), where main() ensures that:

a.The Cortex-M3 system and timers are initialized -SystemInit()

b.the os kernel is initialized for interfacing software to hardware -osKernelInitialize()

c.Creates the threads to execute thread1 and thread2 -Init\_Thread ();

d.Starts the kernel to begin thread switching -osKernelStart(

The Thread\_C thread executes for its round-robin time slice since it is the highest priority. After 15 msec the timer thread forces control to the Thread\_A which has above normal priority and then over to Thread\_D. After 15 msec the timer thread forces control to the Thread\_B which has above normal priority and then over to Thread\_E.