Time Triggered Scheduler functions

void uTTCOS_Dispatch_Threads(void) -- This is the uTTCOS dispatcher' function.
When a task (function) is due to run, uTTCOS_Dispatch_Tasks() will run it. This function must be called (repeatedly) from the main loop.

unsigned char uTTCOS_AddThread(Do_X, delay, period)
Thread is defined as void Do_X(void); Causes a thread(function) to be executed at regular intervals after a fixed delay. Delay and period are measured in tics. Setting period = 0 will make the task a tun_once tasks

unsigned char uTTCOS_AddThread__fI(Do_X, pARG__fI, delay, period)
Causes a thread(function) to be executed at regular intervals after a fixed delay. Thread is defined as void Do_X(int pArg__fI); This means that you can pass an integer to the thread

unsigned char uTTCOS_AddThread__pV(Do_X, pARG__pV, delay, period) -- – Causes a thread(function) to be executed at regular intervals after a fixed delay. Thread is defined as void Do_Xvoid(void *pArg__pV); This means that you can pass an void pointer to the thread -- void pointers can be cast into anything

unsigned char uTTCOS_AddPreEmptiveThread(Fast_Do_X, delay, period) – Causes a thread(function) to be executed at regular intervals after a fixed delay. Thread is defined as void Fast_Do_X(void); This is a high priority thread and MUST be guarenteed to complete execution in one Tick. Only one pre-emptive thread is allowed in uTTCOS

unsigned char uTTCOS_Delete_Thread(const unsigned char TASK_INDEX) -- Removes a thread from the scheduler. Note that this does *not* delete the associated function from memory: it simply means that it is no longer called by the scheduler.

void uTTCOS_Init(void) -- Scheduler initialisation function. Prepares scheduler data structures and sets up timer interrupts at required rate. You must call this function before using the scheduler.

void uTTCOS_Start_Scheduler(void) -- Starts the scheduler, by enabling interrupts. NOTE: Usually called after all regular tasks are added, to keep the tasks synchronised. NOTE: ONLY THE SCHEDULER INTERRUPT SHOULD BE ENABLED!!!

void uTTCOS_Update(void) -- This is the scheduler ISR Callback. It is called at a rate determined by the timer settings in SCH_Init_Scheduler()

void TTCOS_EnterLowPowerMode(void) – Cause processor to enter low power mode

The following functions are processor dependent and need to be built
void SetUpTimerInterrupt(void);
void StartTimerInterrupts(void);
void StopTimerInterrupts(void);
void EnterLowPowerMode(void);