Exp 6::

#include <includes.h>

#include "func.h"

/\*--------------- AAPLICATION STACKS ---------\*/

static OS\_STK AppTaskStartStk[APP\_TASK\_STK\_SIZE];

static OS\_STK AppTaskLED0stk[APP\_TASK\_STK\_SIZE];

static OS\_STK AppTaskLED1stk[APP\_TASK\_STK\_SIZE];

static OS\_STK AppTaskLED2stk[APP\_TASK\_STK\_SIZE];

/\*-------------LOCAL FUNCTION PROTOTYPES--------------\*/

/\*--------------- A PARENT TASK (MAIN TASK) ---------\*/

static void AppTaskStart (void \*p\_arg); /\* Main(Parent)

Task Function \*/

static void AppTaskCreate(void); /\* Separate

Function To Create Child Task(s) \*/

/\*--------------- FOUR CHILDERN TRASKS --------------\*/

static void AppTask1 (void \*p\_arg);

static void AppTask2 (void \*p\_arg);

static void AppTask0 (void \*p\_arg);

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* main()

\*

\* Description : This is the standard entry point for C code. It is assumed

that your code will call

\* main() once you have performed all necessary initialization.

\*

\* Argument(s) : none

\*

\* Return(s) : none

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main (void)

{

BSP\_IntDisAll(); /\* Disable all interrupts until

we are ready to accept them \*/

OSInit(); /\* Initialize "uC/OS-II, The

Real-Time Kernel" \*/

OSTaskCreate(AppTaskStart, /\* Create the

starting task i.e. Main Task \*/

(void \*)0,

(OS\_STK \*)&AppTaskStartStk[APP\_TASK\_STK\_SIZE - 1],

APP\_TASK\_START\_PRIO);

OSStart(); /\* Start

multitasking (i.e. give control to uC/OS-II) \*/

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*

\* AppTaskStart()

\*

\* Description : The startup task. The uC/OS-II ticker should only be

initialize once multitasking starts.

\*

\* Argument(s) : p\_arg Argument passed to 'AppTaskStart()' by

'OSTaskCreate()'.

\*

\* Return(s) : none.

\*

\* Note(s) : (1) The first line of code is used to prevent a compiler

warning because 'p\_arg' is not

\* used. The compiler should not generate any code for this

\* statement.

\*

\* (2) Interrupts are enabled by uCoss-II once the task starts

because

\* main() has disbled it.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*/

static void AppTaskStart (void \*p\_arg)

{

p\_arg = p\_arg; /\*Just to avoid compiler Warning

\*/

BSP\_Init(); /\* Initialize BSP functions

\*/

UartInit(9600); /\*initialise the UART\*/

LEDInit(); /\*Initialize LED \*/

AppTaskCreate(); /\* Create application tasks

(child tasks) \*/

while(DEF\_TRUE)

{

printf(" \r\nMAIN TASK: Created 3 Tasks. Now going to deep sleep...");

printf(" \r\n======================================================");

OSTimeDlyHMSM(1, 0, 0, 0);

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* AppTaskCreate()

\*

\* Description : Create the application tasks.

\*

\* Argument(s) : none.

\*

\* Return(s) : none.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

static void AppTaskCreate (void)

{

/\* Create Task0 \*/

OSTaskCreate(AppTask0,

(void \*)0,

(OS\_STK \*)&AppTaskLED0stk[APP\_TASK\_STK\_SIZE - 1],

APP\_TASK3\_PRIO);

/\* Create Task1 \*/

OSTaskCreate(AppTask1,

(void \*)0,

(OS\_STK \*)&AppTaskLED1stk[APP\_TASK\_STK\_SIZE - 1],

APP\_TASK1\_PRIO);

/\* Create Task2 \*/

OSTaskCreate(AppTask2,

(void \*)0,

(OS\_STK \*)&AppTaskLED2stk[APP\_TASK\_STK\_SIZE - 1],

APP\_TASK2\_PRIO);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* TASK-0 : AppTask0()

\*

\* Description : This task will toggle LED0 i.e. pin P0.12

\*

\* Argument(s) : p\_arg Argument passed to 'AppTask0()' by

'OSTaskCreate()'.

\*

\* Return(s) : none.

\*

\* Note(s) : (1) The first line of code is used to prevent a compiler

warning

\* because 'p\_arg' is not used. The compiler should not

generate

\* any code for this statement.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*/

static void AppTask0 (void \*p\_arg)

{

p\_arg = p\_arg; /\*Just to avoid

compiler Warning \*/

while(DEF\_TRUE)

{

LEDclr(0);

OSTimeDlyHMSM(0, 0, 0, 100); /\* Sleep for a while (500

miliseconds )\*/

LEDset(0);

OSTimeDlyHMSM(0, 0, 0, 100); /\* Sleep for a while (500

miliseconds )\*/

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* TASK-1 : AppTask1()

\*

\* Description : This task will toggle LED1 i.e. pin P0.13

\*

\* Argument(s) : p\_arg Argument passed to 'AppTask1()' by

'OSTaskCreate()'.

\*

\* Return(s) : none.

\*

\* Note(s) : (1) The first line of code is used to prevent a compiler

warning

\* because 'p\_arg' is not used. The compiler should not

generate

\* any code for this statement.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*/

static void AppTask1 (void \*p\_arg)

{

p\_arg = p\_arg; /\* Just to avoid

compiler Warning \*/

while(DEF\_TRUE)

{

LEDclr(1);

OSTimeDlyHMSM(0, 0, 1, 300); /\* Sleep for a while (500

miliseconds )\*/

LEDset(1);

OSTimeDlyHMSM(0, 0, 1, 300); /\* Sleep for a while (500

miliseconds )\*/

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* TASK-2 : AppTask2()

\*

\* Description : This task will toggle LED2 i.e. pin P0.14

\*

\* Argument(s) : p\_arg Argument passed to 'AppTask2()' by

'OSTaskCreate()'.

\*

\* Return(s) : none.

\*

\* Note(s) : (1) The first line of code is used to prevent a compiler

warning

\* because 'p\_arg' is not used. The compiler should not

generate

\* any code for this statement.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*/

static void AppTask2 (void \*p\_arg)

{

p\_arg = p\_arg; /\* Just to avoid

compiler Warning \*/

while(DEF\_TRUE)

{

LEDclr(2);

OSTimeDlyHMSM(0, 0, 3, 700); /\* Sleep for a while (500

miliseconds )\*/

LEDset(2);

OSTimeDlyHMSM(0, 0, 3, 700); /\* Sleep for a while (500

miliseconds )\*/

}

}