Exp8::

#include <includes.h>

#include "func.h"

/\*\*\*\*\*\*\*\*\*\* Define Task Priorities \*\*\*\*\*\*\*\*\*\*\*/

#define APP\_TASK\_START\_PRIO 4

#define APP\_TASK0\_PRIO 5

#define APP\_TASK1\_PRIO 6

#define APP\_TASK2\_PRIO 7

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

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

static OS\_STK AppTask0stk[APP\_TASK\_STK\_SIZE]; /\* Create the

required number of stacks need for every child task\*/

static OS\_STK AppTask1stk[APP\_TASK\_STK\_SIZE];

static OS\_STK AppTask2stk[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) \*/

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

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

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

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

OS\_EVENT \*TxMbox;

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

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

\* 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

\*/

InitLCD(); /\* Initialize LCD \*/

kbdInit(); /\* Initialize Keyboard \*/

ADCInit(); /\* Initialize ADC \*/

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

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

TxMbox = OSMboxCreate((void \*)0); /\* Create Mailbox \*/

AppTaskCreate(); /\* Create application tasks

(child tasks) \*/

while(DEF\_TRUE)

{

printf(" \r\nMAIN TASK: Created 2 Tasks. Now going to deep sleep...");

printf("

\r\n======================================================\n\r");

OSTimeDlyHMSM(1, 0, 0, 0);

}

}

/\*

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

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

\* AppTaskCreate()

\*

\* Description : Create the application tasks.

\*

\* Argument(s) : none.

\*

\* Return(s) : none.

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

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

\*/

static void AppTaskCreate (void)

{

/\* Create User Tasks \*/

OSTaskCreate(AppTask0, //

Name of Task

(void \*)0, //

Pointer to arguments for task execution

(OS\_STK \*)&AppTask0stk[APP\_TASK\_STK\_SIZE - 1], //

Pointer to top-of-stack of the assigned stack

APP\_TASK0\_PRIO ); //

Task Priority

OSTaskCreate(AppTask1, //

Name of Task

(void \*)0, //

Pointer to arguments for task execution

(OS\_STK \*)&AppTask1stk[APP\_TASK\_STK\_SIZE - 1], //

Pointer to top-of-stack of the assigned stack

APP\_TASK1\_PRIO ); //

Task Priority

}

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

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

\* TASK-0 : AppTask0()

\*

\* Description : This task accepts a message from the user upto 16

characters. When ENTER

\* is pressed, the message is posted to the Mailbox.

\*

\* 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)

{

unsigned char temp, msg[20];

unsigned int i;

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

compiler Warning \*/

while(DEF\_TRUE)

{

/\* User Code Here \*/

printf("TASK0: Send a message to TASK0\n\r");

OSTimeDlyHMSM(0,0,0,500);

printf("TASK0: Enter a message from PC Keyboard (max 16 chars): ");

for(i=0;i<16;i++) /\* Accept message from the user \*/

{

temp = UART\_GetChar();

if(temp == 0x0D)

break; /\* if ENTER is pressed i.e. <CR>,

then break out of the for loop \*/

UART\_PutChar(temp); /\* Echo the key pressed \*/

msg[i] = temp; /\* Store the key pressed in the msg

array \*/

}

msg[i] = '\0'; /\* Insert NULL character at the end

of string \*/

while(UART\_GetChar() != 0x0A); /\* wait till <LF> new line

character is recieved after ENTER \*/

OSMboxPost(TxMbox, (void\*)msg); /\* Post the message to the

Mailbox \*/

printf("\n\rTASK0: Message Sent....\n\r");

OSTimeDlyHMSM(0,0,1,0);

}

}

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

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

\* TASK-1 : AppTask1()

\*

\* Description : This task waits for a message to arrive in the mailbox. If

any message is

\* received, it is displayed on the LCD.

\*

\* 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)

{

unsigned char err, \*msg;

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

avoid compiler Warning \*/

while(DEF\_TRUE)

{

/\* User Code Here \*/

printf("TASK1: Waiting for message from Mailbox\n\r");

msg = (unsigned char\*)OSMboxPend(TxMbox,0,&err);

printf("TASK1: Message received....\n\r\n\r");

LCD\_cmd(0x01);

LCD\_display(1,1,msg);

OSTimeDlyHMSM(0,0,1,0);

}

}

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

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

\* TASK-2 : AppTask2()

\*

\* Description :

\*

\* 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)

{

/\* User Code Here \*/

}

}