Exp 7::

#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 \*MsgQ; /\* Message Queue \*/

char \*msg[5] = { "welcome", /\* Create an array of 5 messages \*/

"JSPM",

"RSCOE",

"E&TC",

"Embedded world" };

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

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

\* 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\*/

MsgQ = OSQCreate((void\*\*)&msg,6); /\* Create a message Queue \*/

AppTaskCreate(); /\* Create application tasks

(child tasks) \*/

while(DEF\_TRUE)

{

printf(" \r\nMAIN TASK: Created 3 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 );

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-0 : AppTask0()

\*

\* Description : This task posts a message in the queue after every 2 sec.

\*

\* 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 int i;

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

compiler Warning \*/

while(DEF\_TRUE)

{

/\* User Code Here \*/

for(i=0;i<5;i++)

{

printf("TASK0: Posting message: %s\n\r",msg[i]);

OSQPost(MsgQ,(void\*)msg[i]);

OSTimeDlyHMSM(0,0,2,0);

}

OSTimeDlyHMSM(0,1,0,0);

}

}

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

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

\* TASK-1 : AppTask1()

\*

\* Description : This task accepts a message from the queue after every 3.8

sec.

\*

\* 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 \*/

msg = (unsigned char\*)OSQPend(MsgQ,0,&err);

printf("TASK1: Message received: %s\n\r",msg);

OSTimeDlyHMSM(0,0,3,800);

}

}

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

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

\* 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 \*/

}

}