Ramzi El-abdallah

Lab 7 Code Submission

/\*----------------------------------------------------------------------------

LAB EXERCISE - Real-Time Operating System

----------------------------------------

Integrate functions developed in previous modules and run then concurrently

in the mbed RTOS. The following four threads have to be implemented:

1. Display the temperature on the PC

2. Adjust the frequency of a sound wave using a potentiometer

3. Display an incrementing counter on the PC

4. Blink an LED

GOOD LUCK!

\*----------------------------------------------------------------------------\*/

#include "mbed.h"

#include "rtos.h"

#include "DS1631.h"

#include "pindef.h"

/\*

Define the mutex

Define the PC display and the temperature sensor

Define other inputs and outputs

\*/

//Write your code here

Mutex pc\_mutex;

I2C temp\_sensor(I2C\_SDA, I2C\_SCL);

Serial pc(UART\_TX, UART\_RX);

BusOut LEDS (PB\_10, PB\_4, PB\_5);

AnalogIn pitch(PA\_0);

PwmOut aout(PB\_3);

//Display temperature on the PC

void temp\_thread(void const \*args){

const int temp\_addr = 0x90;

char cmd[] = {0x51, 0xAA};

char read\_temp[2];

while(1){

//write your code here

temp\_sensor.write(temp\_addr, &cmd[0], 1);

wait(0.8);

temp\_sensor.write(temp\_addr, &cmd[1], 1);

wait(0.1);

temp\_sensor.read(temp\_addr, read\_temp, 2);

//Convert temperature to Celsius

float temp = (float((read\_temp[0] << 8) | read\_temp[1]) / 256);

//Print temperature to the serial monitor

pc.baud(9600);

pc\_mutex.lock();

pc.printf("Temperature: %f\n", temp);

wait(1);

pc\_mutex.unlock();

}

}

//Adjust the fequency of the sound wave

void adjust\_frequency(void const \*args){

//write your code here

float vol;

float pit;

float a = 0.003;

float b = 0.000125;

while(1){

//vol = volume.read();

pit = pitch.read();

//for loop that creats the saw tooth wave

//linear function a\*pit+b sets the frequency

//a and b are set to be 8kHz at 0 and 320Hz at 1

for(float i = 0; i < 1; i+=0.05){

aout = i;

aout.period(a\*pit + b);

wait(0.01);

}

}

}

//Blink an LED

void led1\_thread(void const \*args){

//write your code here

while(1){

LEDS = LEDS xor 7;

wait(.2);

}

}

//Display a counter on the PC

void count\_thread(void const \*args){

//write your code here

int count = 0;

while(1){

pc\_mutex.lock();

pc.printf("Count = %d\n", count);

pc\_mutex.unlock();

wait(1);

count++;

}

}

/\*----------------------------------------------------------------------------

MAIN function

\*----------------------------------------------------------------------------\*/

int main(){

/\*

Initialise

Start all threads

\*/

Thread t1(temp\_thread);

Thread t2(adjust\_frequency);

Thread t3(led1\_thread);

Thread t4(count\_thread);

//write your code here

//Wait for timer interrupt

while(1){

\_\_wfi();

}

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ARM University Program Copyright (c) ARM Ltd 2014\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*