UNIVERSITY OF LIVERPOOL

Department of Electrical Engineering and Electronics

ELEC422

Microprocessor Systems Assignment 3 Programming the K64F

Module	ELEC422
Coursework name	Assignment 3
Component weight	40%
Semester	2
HE Level	7
Lab location	EEE-A402 as timetabled, at other times for private study
Work	Individually
Timetabled time	9 hours (3 hours per week – Thursday 9am – noon)
Suggested private study	10 hours including report writing
Assessment method	Individual, formal word-processed reports (Block diagrams / Flow diagrams can be hand drawn and scanned into the report)
Submission format	Online via VITAL
Submission deadline	23:59 on Sunday 12th May 2024 (tbc)
Late submission	Standard university penalty applies
Resit opportunity	August resit period (if total module failed)
Marking policy	Marked and moderated independently
Anonymous marking	No
Feedback	Via printed annotated copy showing corrections
Learning outcomes	LO2: On successful completion of the module students should be able to use a real-time operating system to create a multithreaded program running on a Cortex M series device.

Marking Criteria

Section	Marks available	Indicative characteristics	
		Adequate / pass (50%)	**Very good / Excellent**
Presentation and structure	20%	- Contains cover page information, table of contents, sections with appropriate headings. - Comprehensible language; punctuation, grammar and spelling accurate. - Equations legible, numbered and presented correctly. - Appropriately formatted reference list.	- Appropriate use of technical, mathematic and academic terminology and conventions. - Word processed with consistent formatting. - Pages numbered, figures and tables captioned. - All sections clearly signposted. - Correct cross-referencing (of figures, tables, equations) and citations.
Introduction, Method and Design	40%	- Problem background introduced clearly. - Evidence of a Top Down Design approach - Testing approach introduced	- Appropriate range of references used. - Design decisions justified with alternatives given. - Justification of design choice decisions.
Results	30%	- Results presented with explanations of what they represent.	- Full coverage of all testing and results.
Discussion	10%	- Discussion on what worked and what didn’t.	- Discussion on how the system was fully tested and how the system could be improved.

DEPARTMENT OF ELECTRICAL ENGINEERING AND ELECTRONICS

ELEC422 Microprocessor Systems: Assignment 3: April 2024

40% of the total marks for ELEC422 are for Assignment 3.

This assignment is to allow you to demonstrate your understanding of the features of a Real Time Operating System (RTOS) on the FRDM-K64F Board that you used for the first assignment. You should work through the assignment as a series of stages. Your code should be a series of threads that communicate using the mechanisms of inter-process communication provided by the RTX5 Operating system under the CMSIS-RTOS2 wrapper.

Submission Deadline

Electronic copy: Sunday 12th May 2024 (tbc) @ 11:59pm

Part A – Display Sub-system

Interacting with a user is one of the important features of an embedded system. For Part A you are to create 4 threads and at least 2 interrupt service routines (ISRs).

Thread 1 will manage communications from the PC to the K64F over the Virtual Communications Port (VCP) whilst Thread 2 will manage communications from the K64F to the PC over the VCP. Thread 1 will read text from the keyboard, echo them back to the display via Thread 2 and also decode any valid messages to signal to the other threads. Thread 2 will read messages from a Zero Copy Mailbox where ascii strings of no more than 127 printable characters are passed through the Mail Queue (NB: It is actually the pointer to the memory string that gets passed to the Mailbox – check the notes on Zero Copy Mailboxes)

The two ISRs will monitor the two switches SW2 (PTC6) and SW3 (PTA4) and the 2 threads associated with the ISRs will send appropriate string messages to Thread 2 via the Zero Copy Mailbox when the status of any switch changes e.g. a message “SW2 pressed” or “SW2 released” should be displayed on the PC terminal.

Part B – LED Sub-system

Now add an extra thread, Thread 5, that drives the value of the RGB LEDs when requested to by Thread 1 when it receives the characters ‘r’, ‘R’, ‘g’, ‘G’, ‘b’ and ‘B’ typed by the user. When the character is in lower case it should switch off the LED and when the character is in upper case it should switch on the LED. You should use Flags to provide the mechanism for communications between Thread 1 and Thread 5.

Part C – ADC Sub-system

Now add an extra two threads, Thread 6, which reads the values of the ADC connected to Pot 1 at a periodic rate which can be set by the user typing 1 followed by a space and the sample period in ms (i.e. 1 1000 would correspond to 1Hz sampling – use a default rate of 0.1 Hz until the new rate is received) and Thread 7, which reads the values of the ADC connected to Pot 2 when requested to by Thread 1 when it receives the character “2”. Again you should use Flags to communicate between Thread 1 and Thread 7. Threads 6 and 7 should communicate the read values back to Thread 2 via messages sent to the Mail Queue where the values are then displayed over the VCP. You need to add an appropriate mechanism to stop Thread 6 trying to use the ADC when Thread 7 is using it and vice versa. It is for you to decide a mechanism for Thread 1 to communicate the sample rate to Thread 6. You may want to add extra threads / ISRs to assist Thread 6.

Hint: If you are using global variables for communication you are probably doing it incorrectly.

ELEC422: Assignment 3 Assessment

Part A (40 marks maximum) Note if Part B includes the Code of Part A there is no need to separately document Part A

i) Diagram of inter thread communications

ii) Flow charts of each thread

iii) C Source code of the program.

iv) Explanation of Test Results – Include screen dumps of the “System and Thread Viewer”

Part B (30 marks maximum) – Note if Part C includes the Code of Part B there is no need to separately document Part B (or Part A)

v) Diagram of inter thread communications

vi) Flow charts of each thread

vii) C Source code of the program.

viii) Explanation of Test Results– Include screen dumps of the “System and Thread Viewer”

Part C (30 marks maximum)

ix) Diagram of inter thread communications

x) Flow charts of each thread

xi) C Source code of the program.

xii) Explanation of Test Results– Include screen dumps of the “System and Thread Viewer”