北京郵電大學

Queen Mary University of London

EBU5476

COVID-19 Alternative Assessment (Even Sem – Paper A)

Joint Programme Assessments 2019/20

EBU5476 Microprocessors for Embedded Computing

Answering this paper requires 2 hours; Answers to be submitted within the allocated 6 hours window.

Answer ALL questions

INSTRUCTIONS

1. **You must NOT share any content from this document during the assessment period.**
2. Your answers must be typed, and diagrams or equations must be written clearly and legibly with black or blue color **and in English**.
3. You need to submit your answers BEFORE the allocated deadline.
4. **Read the instructions on the inside cover of the questions sheet.**

Examiners

Dr Matthew Tang, Dr Mona Jaber

Filename: 1920_EBU5476_ALT2020_A

Instructions

This is an open-book exam, which should be completed **within 2 hours**. You MUST submit your answers within 6 hours from the exam being released.

You MUST complete the assessment on your own, without consulting any other person. You MAY NOT check your answers with any other person.

You can refer to textbooks, notes and online materials to facilitate your working, if you provide a direct quote, or copy a diagram or chart, you must cite the source.

Before you start the assessment

1) Read the questions thoroughly and understand them.
2) Ensure you have all the resources you require to complete and upload the final assessment.
3) If you require any assistance, **raise the issue via the messaging section of this assessment on QM-Plus**, immediately.

During the assessment session

1) Use the supplied answer sheet document to enter your answers. Start on a new page for each question. Make sure it is clear which question number you are answering.
2) **Type your answers** in the supplied answer sheet; hand-written equations or sketches can be incorporated into the answer sheet. Please save your work at least every 15 minutes so that you do not risk losing it.
3) When completed answering all questions, performa word count and list the number of words on the answer sheet, then save the file as pdf before uploading, **only pdf will be accepted**, any other file format will not be accepted.
4) Your submission must be your own work, and you must ensure that you do not break any of the rules in the Academic Misconduct Policy.

Submitting the Assessment

1) You will have 6 hours from the start of the scheduled assessment time – do not leave submissions too close to the deadline. NO late submission will be accepted, no exceptions.
2) Make sure you upload and submit the final version before the deadline.
3) Please be aware that submissions will be subject to review, including but not limited to plagiarism detection software.

If you have any problems relating to access or submitting during the exam period, please contact the email (it-issues@qmbupt.org), state the module code in the subject, and clearly state your name and student ID and any issues you are experiencing. You must use either @qmul.ac.uk or @bupt.edu.cn email address. Requests from external email addresses will not be processed.

Question 1

a) The ARM Cortex-M4 is a high-performance embedded solution developed to optimize for low-cost and energy-efficient embedded systems.
Word limit: 200 words

[10 marks]

i) In a typical low-cost embedded system, which components are commonly used for collecting, processing and distributing data? Provide example of an embedded system using these components and explain their functionality in your proposed system.

(5 marks)

ii) Why do you think APSR is necessary in the Cortex-M4 microprocessor? Provide an example of an application using at least two of the condition code flag bits from APSR.

(5 marks)

b) The following shows a simple C program snippet written for Cortex-M series microprocessors:

```
1  int disp(int x, int y){
2      // your assembly code
3  }
4
5  void main(void){
6      const int x[4] = {0,  2, 4, -1};
7      const int y[4] = {0, -1, 3,  3};
8      int d[4];
9      for (int i = 0; i < 4; i++)
10         d[i] = disp(x[i], y[i])
11     ...
12 }
```

The function `disp()` calculates:

$$\Delta = (x-a)^2 + (y-b)^2$$

where a and b are the last two digits of your **QMUL** student ID (e.g. 170123456, then a = 5 and b = 6).

Complete the assembly code for the function `disp()` following AAPCS with **no more than 30 lines/instructions**.

Then discuss (**in more than 100 words**) the ideal memory location to store the arrays of variables (`x[]`, `y[]` and `d[]`) declared.

[15 marks]

Question 2

a) The Cortex-M4 CPU performs certain processing automatically to respond to an exception request. Answer the following questions regarding exceptions and interrupts in Cortex-M4 processor.
Word limit: 200 words

[10 marks]

i) Suppose a Nested Vectored Interrupt Controller (NVIC) is handling a 'Non-Maskable Interrupt' and a new exception is requested during the execution. Explain what happens if the new exception is a **HardFault** exception?

(4 marks)

ii) If a CPU generates an `EXP_RETURN` value as 0xFFFF_FFED, explain briefly how the CPU determines which SP the registers to be restored from, and which mode will that return to.

(3 marks)

iii) The figure below represents the contents of some registers and their corresponding addresses (indicated on the left). Suppose an interrupt occurs and the Stack Pointer starts pushing from xPSR up to R0 of the Stack Frame to the memory, as indicated in the figure.

What will be the SP value upon entering the ISR and the address in the blank box?

Give your answers in hexadecimal values and briefly explain the logic behind your answers.

(3 marks)

	200562CB	00000000	000000E2
0×30000FC8:	200562CB	00000824	00000000
0×????????:	F929F000	0000008C	0FFFF0AA
	00000A0B	00000A80	21000000
			↑ *Saved xPSR*

b) Study the recursive function `f()` as coded below:

```
1 int f(int n){
2     if (n <= 1)
3         return 1;
4     return (n * f(n - 1));
5 }
```

Explain the role and the importance of the stack in executing this function.

Thus estimate the memory space requirement (in terms of bytes) if the function is called by `f(10)`.

Explain briefly how the condition on line 2 can be translated (or compiled) as ARM instructions.

Word limit: 200 words

[15 marks]

Question 3

a) Cortex-M series microprocessors are equipped with many useful timer peripherals (e.g. pulse width modulation timer, low-power timer, real-time clock, etc.)

Choose one type of timer peripherals. Discuss and explain an application of that in a certain consumer product.

You should highlight the importance of the chosen timer peripheral and justify its necessity in the system.

Word limit: 150 words

[10 marks]

b) Using GPIOs ports of an STM32F401 MCU, design a system that takes a number A as an input and displays the sum $A + B$ in binary format as output using LEDs. The number B is computed based on the last digit of your QMUL ID number Q where $B = \min(Q, 8)$ in binary format. The number A is also in binary format and is determined by THREE switches that represent the least significant bits $A = \{0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111\}$. For example, if $A = 0111$ and $Q = 9$, then $B = 1000$ and the output is 1111 and four LEDs should be lit. A push button is used to control the summation; only when the button is pressed and then released, the LEDs are activated. Provide a diagram that describes your system, the C code of the main function, and enter your results in the following table:

QMUL ID last digit	Q = ?
$A = Q \bmod 2$	$A + B$ = ?
$A = Q \bmod 3$	$A + B$ = ?
$A = Q \bmod 5$	$A + B$ = ?

You can use the functions `gpio_get()`, `gpio_set()` and `gpio_set_mode()` from the lecture notes in your answer. **PLEASE CITE ANY REFERENCE THAT YOU USE**, for example, cite "lecture notes JPESD_08-GPIO slide 28" if you decide to use `gpio_set()` function.

Word limit: 200 words (including citations/references)

[15 marks]

Question 4

a) You are given a list of devices to be connected with a Cortex-M microprocessor:

Device	I^2C address
Bluetooth module	0x2D
LCD Display	0x37
Accelerometer	0x1E

Use a labelled diagram to that illustrates the necessary electrical connections between the microprocessor and **ANY TWO** of the devices from the above table.

Also give an example of either a **master-transmitter** or **master-receiver** transfer that can happen in your system (as in the diagram). Your answer should include a list of **ALL** data bits in the transfer.

Word limit: 100 words (excluding diagram)

[10 marks]

b) This question involves flying drones that are controlled remotely from the ground control station (GCS). The drones are equipped with an ASCII Serial Data NMEA-0183 driver over a RS232 serial communication channel that uses radio frequency transmission/reception to communicate with the GCS. Two channels are established between the drone and the UAV: control channel and video transmission channel. The control channel is always ON and needs to be robust despite noise and fluctuations in the wireless medium. The video channel is on-demand when the drones is asked to transmit video in real-time.

You are asked to setup the serial communication channels between the drone and the GCS by setting up the parameters of the UART device assuming an STM32F401 is used in both drone and GCS. The MCU clock is 84 MHz and you are asked to design the control channel communication in the most robust and resilient way possible and the video transmission channel in the highest data bit rate possible. Please note the difference between the symbol rate and data rate in your design and consider parameters such as data size, checksum bit, oversampling, and others.

Explain and justify your design and **PLEASE CITE ANY REFERENCE THAT YOU US**, for example "lecture notes JPESD_10-SerialCommunication slide 12".

Due to fluctuations in the wireless channel, the GCS or the drone may decide to modify the baud rate of the video transmission channel. A control message is sent on the control channel in the form of an ASCII command string "`AT+BAUDX`".

Write a short C program to poll an ARM UART port for an ASCII command string "`AT+BAUDX`" and change the baud rate according to the table below. Use the last digit of your QMUL ID number Q to calculate $X = Q \bmod 5$. Your program should respond with an ASCII message "`OK`" followed by the actual Baud rate in bps, example "`OK9600`", if the command is completed. You can use the driver functions `uart_init()`, `uart_rx()` and `uart_tx()`.

X	Baud Rate
0	9.6 Kbps
1	38.391 Kbps
2	461.538 Kbps
3	3.652 Mbps
4	7 Mbps

Word limit: 200 words (including citations/references)

[15 marks]

Appendix I: ARM Thumb-2 Instruction Excerpt

Operation	Assembler	Operation	Assembler
Move	`MOV Rd, <Operand2>`	Move NOT	`MVN Rd, <Operand2>`
Move from PSR	`MRS Rd, <PSR>`	Register to PSR	`MSR <PSR>, Rm`
Immediate to PSR	`MSR <PSR> #immediate`		
Load (pre-index)	`LDR Rd, [Rn, #<offset>]`	Store (pre-index)	`STR Rd, [Rn, #<offset>]`
Load (post-index)	`LDR Rd, [Rn], #<offset>`	Store (post-index)	`STR Rd, [Rn], #<offset>`
Add	`ADD Rd, Rn, <Operand2>`	Add with carry	`ADC Rd, Rn, <Operand2>`
Subtract	`SUB Rd, Rn, <Operand2>`	Subtract with carry	`SBC Rd, Rn, <Operand2>`
Multiply	`MUL Rd, Rm, Rs`		
Divide, unsigned	`UDIV Rd, Rm, Rs`	Divide, signed	`SDIV Rd, Rm, Rs`
Compare	`CMP Rn, <Operand2>`	Compare, negative	`CMN Rn, <Operand2>`
Test	`TST Rn, <Operand2>`	Test equivalence	`TEQ Rn, <Operand2>`
Branch	`B <label>`	Branch with link	`BL <label>`
Branch and exchange	`BX Rm`	Branch with link and exchange	`BLX <label>`
Push	`PUSH <reglist>`	Pop	`POP <reglist>`

– End of Appendix I –

END OF PAPER

DO NOT WRITE ON THIS PAGE