UTHM
Universiti Tun Hussein Onn Malaysia

UNIVERSITI TUN HUSSEIN ONN MALAYSIA

FINAL EXAMINATION
SEMESTER II
SESSION 2015/2016

COURSE NAME	:	REAL-TIME EMBEDDED SYSTEM
COURSE CODE	:	BEH 30802
PROGRAMME CODE	:	BEJ
EXAMINATION DATE	:	JUNE / JULY 2016
DURATION	:	2 HOURS 30 MINUTES
INSTRUCTION	:	ANSWER **ALL** QUESTIONS

THIS QUESTION PAPER CONSISTS OF **SIX (6)** PAGES

Q1 (a) List **two (2)** situations where a real-time operating systems is needed in embedded system.

(2 marks)

(b) Identify **four (4)** advantages of using real-time operating system (RTOS) over general purpose operating system (GPOS).

(4 marks)

(c) Recommend suitable of real-time system types for the following systems with a short justification for each of them.

(i) Microwave oven.

(2 marks)

(ii) Fridge-freezer.

(2 marks)

Q2 (a) Explain how to use an external intterupt pin in Arduino Uno system.

(2 marks)

(b) An Arduino-based mobile robot is designed to navigate for following a white line on black background. Its system architecture and complete schematic are given in **Figure Q2(b)**.

(i) Investigate the LED connection mode and logic state for turning off LED at pin D9.

(2 marks)

(ii) The user button is connected to pin D8 and has a problem of floating state during inactive (i.e. button not pressed). Find **two (2)** solutions to this problem.

(4 marks)

(iii) Propose a technique how to measure a speed (in pulse per second) of motor by using an incremental encoder.

(4 marks)

(iv) Assume that left motor (ML) and right motor (MR) have the same specification and performance. Complete the following subroutine for controlling speed and direction of both motors with correct syntax and code statement.

```
void robotMOVE(your parameter for controlling speed and direction)
{
        //your code for controlling speed and direction of each motor
}
```

(4 marks)

Q3 (a) Explain the following terminologies in real time operating system.

(i) Counting Semaphore

(2 marks)

(ii) Mutex

(2 marks)

(b) Show how a task can change their state in ChibiOS scheduler.

(6 marks)

Q4 (a) Explain the following terminologies in temporal scope of a task with an illustration for each.

(i) Completion time

(2 marks)

(ii) Elapse time

(2 marks)

(b) Assume the Arduino-based mobile robot as in **Q2(b)** has a tick period of 10 ms to keep track of time and consists of four independent tasks. The task priorities, periods and CPU (execution) times for the four-task system are tabulated in **Table 1**.

Table 1: Task Execution

Task	Priority No. (i)	Period (P_i)	CPU time (C_i)
A	1	10 ms	5 ms
B	2	50 ms	1 ms
C	3	20 ms	10 ms
D	4	30 ms	5 ms

(i) Plot a task activition diagram for time, $t = 0$ ms to $t = 100$ ms for all tasks.

(6 marks)

(ii) Analyse the start delay, elapse time, deadline and completion time by showing the result in a table for the four-task system.

(6 marks)

(iii) Reconstruct an appropriate priority assignment for each of the four tasks according to the rate monotonic scheduling (RMS) algorithm.

(3 marks)

(iv) Analyse the ability of each task to meet its deadline based on RMS calculation.

(8 marks)

(v) Give an opinion on the result of **Q4(b)(i)** and **Q4(b)(iv)**, which priority assignment is suitable for implementing in the mobile robot system.

(2 marks)

Q5 The operation of Arduino-based mobile robot system as in **Q2(b)** is described in **Table 2**.

Table 2: Task Operation

<table>
<tr><th>Task</th><th>Description</th><th>Period</th></tr>
<tr><td>A</td><td>Read line sensors and save ADC result data into 'Line1', 'Line2' and 'Line3' global variables.</td><td>10 ms</td></tr>
<tr><td>B</td><td>Toggle LED</td><td>50 ms</td></tr>
<tr><td>C</td><td>Control the robot movement based on ADC result data from Task A.
<table>
<tr><th>Line1</th><th>Line2</th><th>Line3</th><th colspan="2">Robot Movement Control</th></tr>
<tr><th>Pin A1</th><th>Pin A2</th><th>Pin A3</th><th>ML</th><th>MR</th></tr>
<tr><td>0</td><td>1</td><td>0</td><td>80% DC</td><td>80% DC</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>30% DC</td><td>80% DC</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>80% DC</td><td>30% DC</td></tr>
<tr><td colspan="5">Note:
1. '1' means white line detected (ADC data≤50).
2. '0' means black background detected (ADC data ≥ 1000).
3. 'DC' means duty cycle.</td></tr>
</table></td><td>20 ms</td></tr>
<tr><td>D</td><td>Send a robot status to computer through UART interface (9600 bps).</td><td>30 ms</td></tr>
</table>

(a) Inspect the syntax error and incorrect code in the following *setup( )* and *chSetup( )* functions. The priority is based on **Q4(b)(v)**.

```
void setup( )
{
   pinMode(9,INPUT); pinMode(4,OUTPUT); pinMode(7,INPUT)      //Configure Input and Output pins
   pinMode(8,OUTPUT); digitalWrite(8,HIGH)
   Serial.begin(115200)                                       //Configure serial baud rate
   digitalWrite(9,HIGH)                                       //LED is turned OFF
   chSetup( );                                                //start ChibiOS
   while(FALSE)                                               //loop here forever
}

void chSetup( )
{
   chThdCreateDynamic(waTaskA, sizeof(TaskA), NORMALPRIO,     //Create Task A
   TaskA, NULL)
   chThdCreateDynamic(waTaskB, sizeof(TaskB), NORMALPRIO,     //Create Task B
   TaskB, NULL)
   chThdCreateDynamic(waTaskC, sizeof(TaskC), NORMALPRIO,     //Create Task C
   TaskC, NULL)
   chThdCreateDynamic(waTaskD, sizeof(TaskD), NORMALPRIO,     //Create Task D
   TaskD, NULL)
}
```

(7 marks)

(b) Point out the standard of ChibiOS/RT task format and relevant C language statement for Task C subroutine. Use the same *robotMOVE* function as in **Q2(b)(iv)** for controlling the robot. Assume the working area size of Task C is 128 bytes.

(10 marks)

(c) Construct a flow-chart for task A and B.

(6 marks)

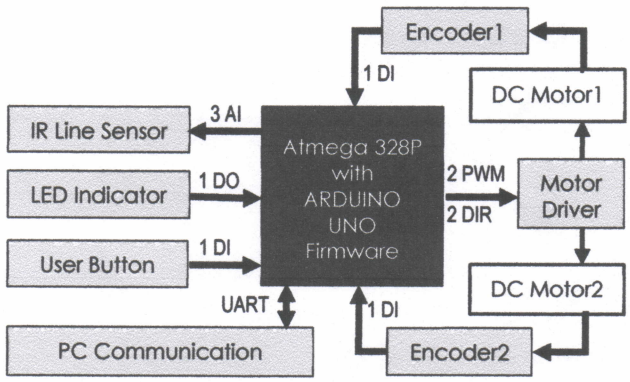
(d) Write a complete programming code for a subroutine of task A and B by referring to the flow-chart in **Q5(c)**. Assume the working area size for both tasks is 100 bytes.

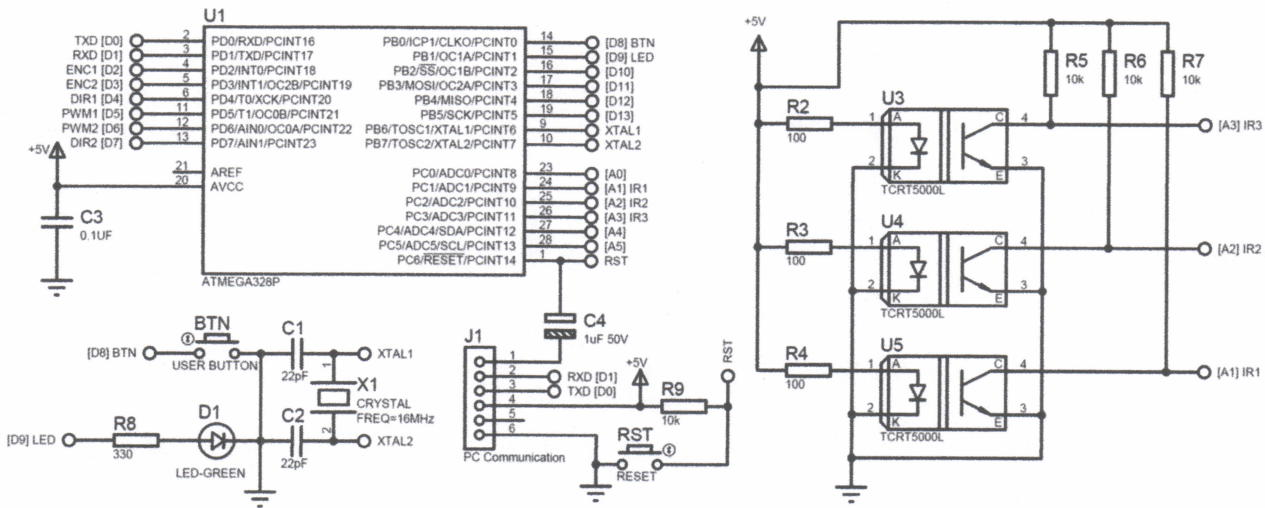
(12 marks)

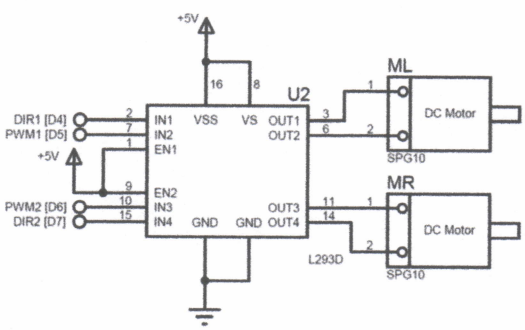
- END OF QUESTIONS –

FINAL EXAMINATION

SEMESTER / SESSION	: SEM II / 2015/2016	PROGRAMME	: BEJ
COURSE	: REAL TIME EMBEDDED SYSTEM	COURSE CODE	: BEH30802







Component	Model	Arduino Pin	Label	Peripheral
Infra-red Line Sensor	Vishay Reflective Optical Sensor (TCRT5000L)	A1	IR1	Analog Input (AI)
		A2	IR2	
		A3	IR3	
PC Communication	FTDI USB to UART or Bluefruit EZ-Link	D0	RXD	UART
		D1	TXD	
Encoder1A	MiniQ chassis Encoder (SKU: SEN0116)	D2	ENC1	Interrupt INT0
Encoder2B		D3	ENC2	Interrupt INT1
Motor 1 (ML)	1. Motor Driver (L293) 2. DC Brushed Motor (6V Micro Metal Gear)	D4	DIR1	Digital Output (DO)
		D5	PWM1	Pulse Width Modulation (PWM)
Motor 2 (MR)		D6	PWM2	
		D7	DIR2	Digital Output (DO)
User Button	Micro-switch	D8	BTN	Digital Input (DI)
LED Indicator	3mm Green LED	D9	LED	Digital Output (DO)

Figure Q2(b)