Đề kiểm tra - Học kỳ I (2010-2011)

Môn: **Các hệ thời gian thực** (lớp CĐT07)

Ngày kiểm tra: 06/11/2010

Thời gian: 50 phút

Ghi chú: Đề kiểm tra được in hai mặt. Sinh viên được phép sử dụng tài liệu. Đề kiểm tra được đánh giá theo thang điểm 20, sau đó được chia đôi để quy sang thang điểm 10.

1. Giải thích ngắn gọn tại sao bản chất của hệ thời gian thực là "multitask"? *(4 điểm)*
2. Động cơ điện một chiều không chổi than (Brushless DC electric Motor) được điều khiển dựa trên nguyên tắc điều khiển độ rộng xung của dòng điện đến động cơ. Hệ thống điều khiển ổn định vòng quay cho một động cơ điện không chổi than theo một vòng quay mong muốn gồm các thành phần sau: encoder để đếm số vòng quay hiện tại, vi xử lý điều khiển độ rộng xung, Electronic Speed Control để điều khiển động cơ, pin LiPo cấp nguồn. Người điều khiển nhập vòng quay mong muốn thông qua một giao diện điều khiển từ máy tính. Độ rộng xung của dòng điện đến động cơ được xác định dựa vào chênh lệch giữa vòng quay ở thời điểm hiện tại với vòng quay mong muốn. Áp dụng phương pháp phân tích SA-RT (Structured Analysis for Real-Time Systems) hãy cho biết:
   a. Biểu đồ bối cảnh dữ liệu *(2 điểm)*
   b. Biểu đồ dòng dữ liệu/dòng điều khiển *(2 điểm)*
   c. Biểu đồ chuyển đổi trạng thái *(3 điểm)*
3. Cho đoạn code Ada sau:

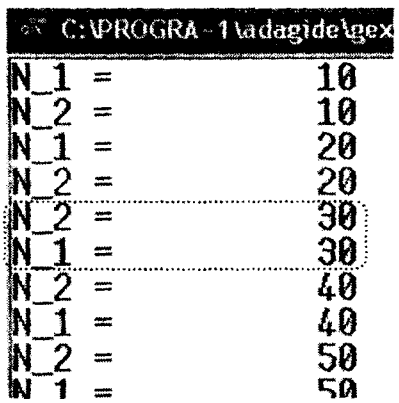
```
-- Real_time_tasks.ads
package Real_Time_Tasks is
   task T1;
   task T2;
end Real_Time_Tasks;
```

```
-- Real_time_tasks.adb
with Ada.Text_Io; use Ada.Text_Io;
with Ada.Integer_Text_Io; use Ada.Integer_Text_Io;
with Ada.Real_Time; use Ada.Real_Time;
package body Real_Time_Tasks is
   task body T1 is
      N_1: Integer := 0;
   begin
      loop
        for I in 1..10 loop
              N_1 := N_1 + 1;
        end loop;
        Put("N_1 = ");
        Put(N_1);
        Put_Line("");
        delay 3.0;
      end loop;
   end T1;
   task body T2 is
      N_2: Integer := 0;
      Next_Release: Ada.Real_Time.Time;
      Period: Integer := 3000;
   begin
      Next_Release := Ada.Real_Time.Clock;
      loop
```

```
            for I in 1..10 loop
                  N_2 := N_2 + 1;
            end loop;
            Put("N_2 = ");
            Put(N_2);
            Put_Line("");
            Next_Release:=Next_Release+Ada.Real_Time.Milliseconds(Period);
            delay until Next_Release;
         end loop;
      end T2;
   end Real_Time_Tasks;
```

a. Hãy cho biết đoạn code trên nói lên điều gì? Hãy biểu diễn đoạn code trên theo biểu đồ cấu trúc task DARTS? *(6 điểm)*

b. Khi chạy đoạn code trên ta thu được kết quả dưới đây



Hãy giải thích tại sao có sự khác biệt về thứ tự xuất của giá trị N_1 và N_2 kể từ giây thứ 9 của chương trình? *(3 điểm)*

Chúc anh/chị làm bài kiểm tra đạt kết quả tốt!

Ngày 01/11/2010,

Người ra đề

Ngô Khánh Hiếu

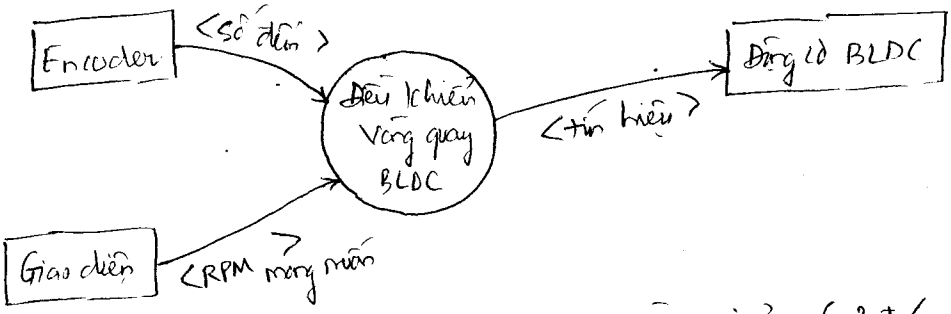
Đáp án Ktra Học kỳ I (2010 - 2011)
Môn: Các hệ thời gian thực
Ngày ktra: 06/11/2010

Ng) Khánh Thiều

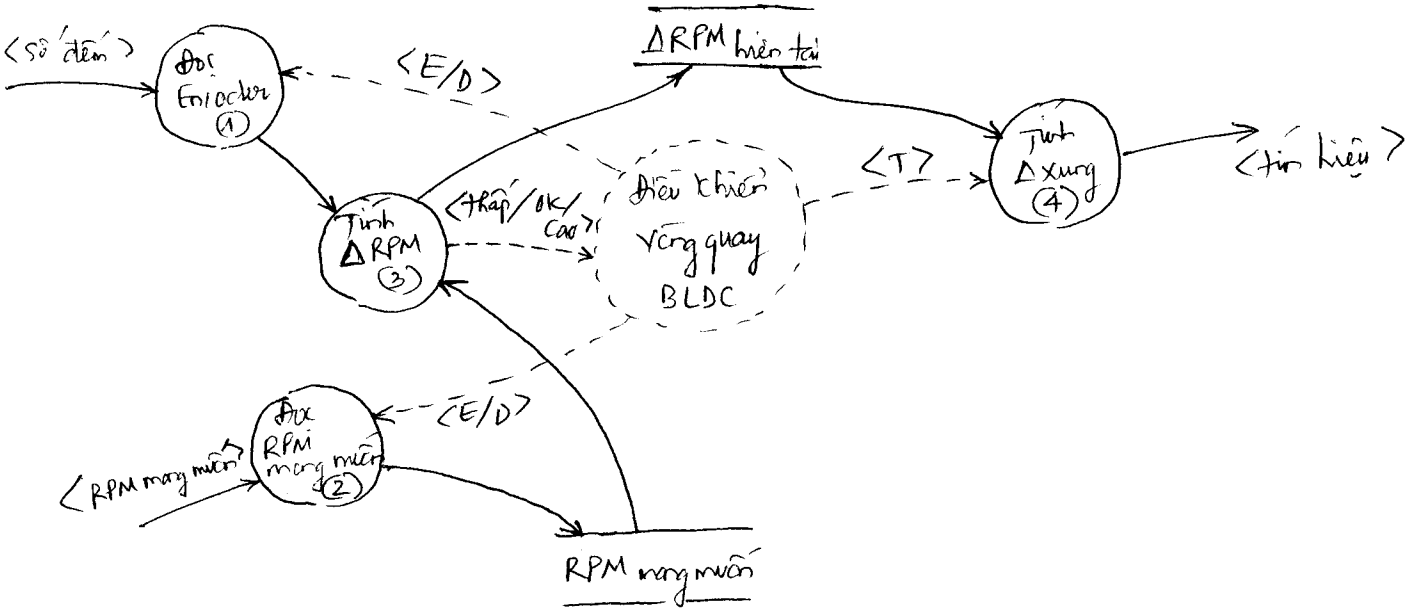
Câu 1 (4 điểm/20)

Multitask
↳ (a) Xử lý đồng thời các dữ kiện điều khiển đến từ môi trường điều khiển ⇒ phản ứng kịp các thay đổi (nếu có) từ môi trường điều khiển
↳ (b) Do bản chất của hệ điều khiển độc lập với sự vận hành của môi trường điều khiển → multitask giúp đảm bảo khả năng cảm nhận môi trường điều khiển của hệ điều khiển.

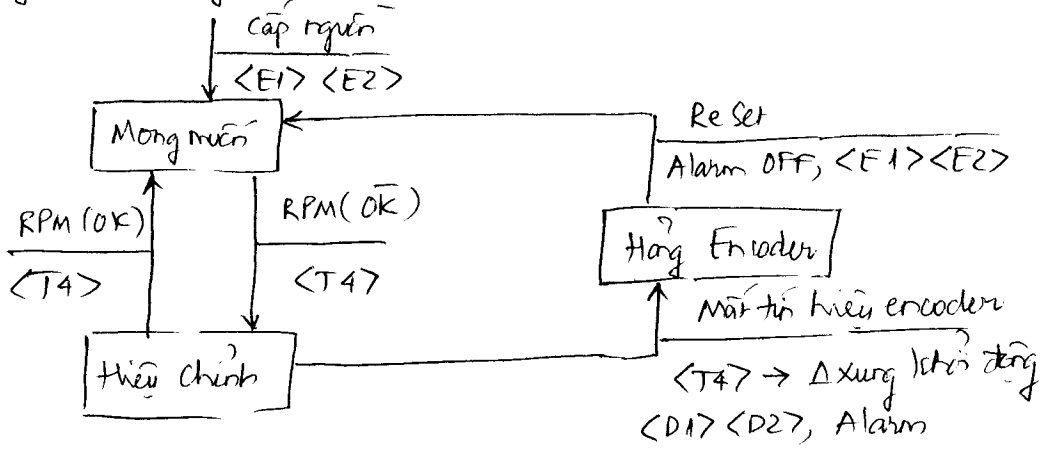
Câu 2 (2đ/20)
a/ Biểu đồ bối cảnh dữ liệu.



b/ Biểu đồ ~~chuyển đổi~~ dòng dữ liệu / dòng điều khiển (2đ/20)



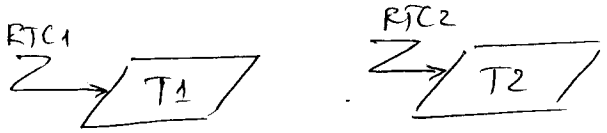
2. c/ Biểu đồ chuyển đổi trạng thái (3đ/20)



Câu 3

a/ (6đ/20) Đoạn code Ada thể hiện 2 task có chu kỳ, mỗi task ở mỗi vòng lặp của nó sẽ xuất ra màn hình 1 giá trị nguyên N-1/N-2 có giá trị lần lượt là 10, 20, 30, ...

Mô hình cấu trúc task DARTS của đoạn code này là:



b/ (3đ/20)

Sự khác biệt về thứ tự xuất giá trị N-1 và N-2 xảy ra là do task 1 dùng hàm [Delay 3.0;] để xác định chu kỳ; còn task 2 thì dùng hàm [Delay Until Next_Release;] để xác định chu kỳ

Bản chất của delay và delay until là khác nhau: Delay xác lập khoảng thời gian chờ kể từ thời điểm kích hoạt hàm này; Delay Until xác lập thời điểm kích hoạt task ở mỗi chu kỳ. Do đó

$$\begin{cases} RTC1 > 3.0 \text{ giây} \\ RTC2 = 3.0 \text{ giây} \end{cases}$$

Chính sự trễ của RTC 1 dẫn đến sự thay đổi trật tự tính giữa nó (task 1) và task 2.

-- Hết --