Université d'Ottawa
Faculté de génie

École de science informatique
et de génie électrique

uOttawa

L'Université canadienne
Canada's university

University of Ottawa
Faculty of Engineering

School of Electrical Engineering
and Computer Science

COURSE: CEG4166/CSI4141 Real Time Systems Design
PROFESSOR: Gilbert Arbez
SEMESTER: Winter 2017
DATE: **April 26, 2017**
TIME: **9h30-12h30**

FINAL EXAMINATION
Evaluation Criteria

Name and Student Number: _____/_____

There are 3 parts in this examination.

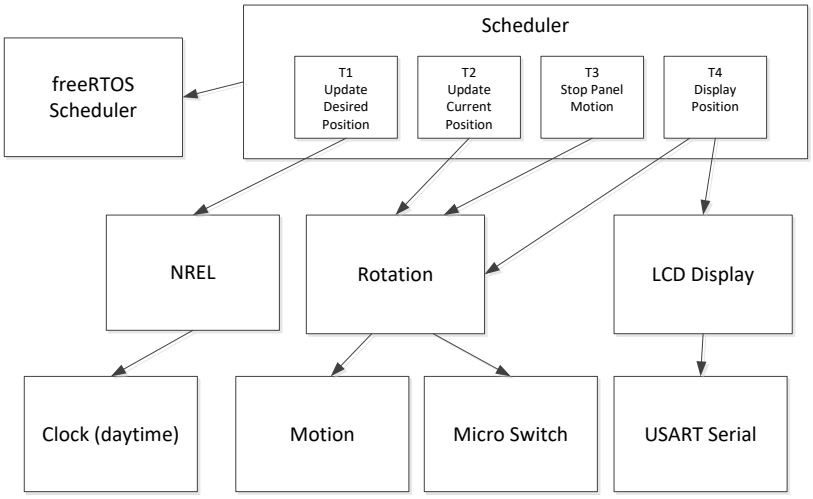
Question 1		10 marks	
Question 2		15 marks	
Question 3		15 marks	
Total		40 marks	

Answer questions in the provided examination booklets.

This is an open book exam.

Total number of pages: 4

Question 1 – Software Modules (10 marks)



Sprint Modules:

- Scheduler Module (can be included in the main module or separate)
    - Contains separate tasks that calls other modules. Not required to show them in Parnas diagram.
- Rotation Module: for reading the encoder and maintaining the position of the panel (in degrees). Provides functions to start and stop the rotation. This module can be separated into 2 modules, one for the rotation of the panel, and a second for updating the position of the panel.
- Micro-Switch module: (can be included in the Rotation module).
- LCD Display Module: Similar design to course project.
- NREL Module: Gives the position to point the panel in degrees between daybreak and nightfall. Defines position as 90^{o} when sun is set.

Given Modules:

- freeRTOS Task Module: for creation of a task to support the cyclic scheduler
- CEG4166 usartserial.c Module: To support communication with the LCD serial display.
- CEG4166 motion.c Module: For manipulating the servo-motors and reading from the encoder.
- Clock: Maintains and provide the time of day.

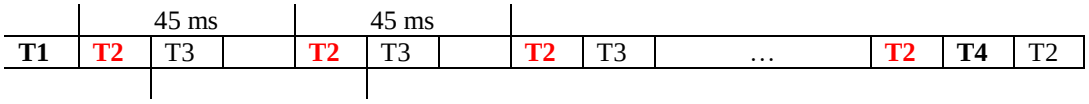
Diagram: Must be shown to relate the various modules. In its simplest form simple boxes relating which modules calls which modules. Can use a UML form that may show global variables and entry point functions.
Each module should have a brief description.

	Description	Mark
B	Base mark • Diagram of modules • Provide brief description of all modules	7
M1	Complete modules (see above)	1
M2	Separation of concerns • NREL module should be separated from other modules (complex algorithm) • Day/Timer module should be separate.	1
M3	Showing all relationships between modules	1

Question 2 – Scheduler Design (15 marks total)

Provide the design of the module responsible for implementing the cyclic scheduler.
Background/Algorithm

- Quick introduction to cyclic scheduler
- Describes how the major and minor cycles are defined. May be formal or informal.
  - Major cycle should be no more than 60 sec – since every minute the position of the panel needs to be adjusted during the day.
  - Minor cycles depend mainly on the periods to update the servomotor and to read the encoder which are both 50 ms. Thus for the tasks below, the minor cycle should be at most 16.667 ms so that Task 2 and Task 3 periods of 50 ms are respected. 15 ms is selected with 400 minor cycles/major cycle which gives a major cycle of 6 sec. The reason for using a shorter major cycle time is to keep the number of minor cycles smaller and thus a shorter task table.
  - Display is set also every 6 seconds, at the end of the major cycle.
- Four tasks are required:
  - Task1 (T1) – Update Desired Position: Calls the function in the NREL module to get the desired position of the panel. If the actual position of the panel does not correspond to the desired position (call the function in the Rotation module to get current position): if the desired position is 90^0 call the function in the Rotation module to start moving the panel east is called, otherwise call function in the Rotation Module to move the panel west. Period = 60 sec.
  - Task 2 (T2) – Update Current Position: Call function in the Rotation module to update and get the current position of the panel. Period = 50 ms.
  - Task 3 (T3) – Stop the Panel Motion: Stop motor if either micro-switch is closed and panel is moving east or if current position exceeds the desired position and the panel is moving west. Do nothing if the panel is stopped. Period = 50 ms.
  - Task 4 (T4) – Update the LCD Display: Update the position of the panel on the LCD display. Period = 6 sec (here the position is updated once per major cycle time, since the position is check once each major cycle time).
  - The following shows a schedule with a minor cycle of 15 ms and a major cycle of 6 s.



- Describes data structures used for implementing the scheduler (table with pointers to functions):
  - desiredPosition: value in degrees that gives the desiredPosition
- Describes the tasks, either local tasks or tasks defined in other modules.
- Tasks are local to the module: important that they make calls to function in other modules – that is, details of the processing should be defined in other modules.
- API: 1) Initialisation function 2) freeRTOS Scheduler task function that may be called by main() function or a function that invokes the freeRTOS Scheduler task.

	Description	Mark
B	Base mark • Introduces the rudiments of cyclic scheduling • Provides all structures required for cyclic scheduling • Describes tasks used in the cyclic scheduling • Provides analysis of cyclic scheduling	10
M1	Proper definition of minor and major cycle (see above)	2
M2	Proper definition of 4 tasks as shown above.	2
M3	Deals with both day tracking (with encoder) and night tracking (move to east position at night)	1

Question 3 – Designing of Positioning Functionality (15 marks total)

Provide the design of one or more modules that interacts with the servo motor and encoder to rotate the panel to the desired position.

This can be designed as a single module or as 2 separate modules – one for the servo-motor, and one for the encoder.

Background/Algorithm

- Hardware design provides description of hardware to be used – no need to include in this section.
- Describe global variables used in the module: e.g.
    - currentPosition: position which gives the position of the panel in cardinal degrees (90^o for east),
    - servomotorState (MOVING_WEST, MOVING_EST, STOPPED), etc.
- Give an overall description of the modules logic: e.g. – the servomotor must be moved to the east position at the end of the day using the micro-switch, i.e. when the direction of the panel is 90^0 (this value is returned by NREL module).
- API: require the following function entry points (this is not the only possible design). In this design task functions are defined within the scheduler module and these functions are called by them.
    - initialisation function() – to initialise global variables.
    - updateCurrentPosition() function: Updates the position of the panel in degrees with a call to the Motion module, when an edge is reported by the motion module, the position is incremented by 0.7^o. Note that the panel is moved only from east to west during the daylight hours. If the micro-switch is closed, the position is set to 90^o.
    - getCurrentPosition() Function: Returns the position of the panel in degrees.
    - startMoving(desiredPosition) Function: Starts the servo-motor moving east if desiredPosition is 90^o (east) and currentPosition is not east. Set servomotorState to MOVING_EAST. Start the servo-motor moving west if the desired position is greater than the current position. Set servomotorState to MOVING_WEST. Otherwise – do nothing.
    - stopMoving(currentPosition) Function: To stop the servo-motor. Stop the motor if moving east (servomotorState is MOVING_EAST) and micro-switch is closed or if moving west (servomotorState is MOVING_WEST) and desiredPosition greater than position. Sets servermotorSate to STOPPED. If motor is stopped (servomotorState is STOPPED), do nothing.

	Description	**Mark**
B	Base mark • Provides background and description of logic of the module • Introduces data structures used in the module • Logic should be consistent with timing constraints and support of tasks developed in question 2. • Well defined API	10
M1	Proper encoder functions to determine current position of the panel	2
M2	Proper servomotor functions to start and stop the servomotor – these functions may run in different tasks.	2
M3	Use of micro-switch to move the panel to the east at the end of each day.	1