



Low Level C-programming and microcomputer architecture

Task 14 RTK Application

Content

1 Introduction	1
2 Requirement specification	2

1 Introduction

a new procurement that requires knowledge of real-time cores, task communication, different I/O and a good insight into how software works in a built-in system. Since the designer possesses these skills, they must carry out the project.




The system to be built will be used at a trade fair as a demonstration of the card and of the RTOS.



2 Requirement specification

Questions regarding the specification should be directed to the customer. The following must be completed in order for the task to be considered completed:

Table 1. Requirement specification from customer.

Requirement ID	Description	Performed Yes/No				
Design requirements						
1	The system should be connected to a VGA monitor. When the system is started, a welcome page (including the name of the designer) should appear on the screen. The welcome page should appear until the user presses a button.					
2	<p>The program will divide the VGA screen into four different tasks, each responsible for a square on the screen (see next figure).</p> <ul style="list-style-type: none">• Task_ACC – reads X, Y, Z accelerator values and writes once a second values (data) on the screen.• Task_ACC_Filter - retrieves the information (global data) from the previous task and prints averages (X, Y, Z) of the last 10 samples.• Task_time - prints the time the system has been working, i.e. from the moment the button was pressed on the welcome screen.• Task_plot - Plots a graph showing changes on the Z-axis (from global data) over time. <div><table><tr><td>Task_ACC - skriver ut accelerometervärdena, en gång i sekunden X = +128 Y = +128 Z = +000</td><td>Task_ACC_filter - skriver ut medelvärdena på 10 sampel X = +096 Y = +130 Z = -002</td></tr><tr><td>Task_time - skriver ut tiden från programstart 00:33</td><td>Task_plot - ritar graf (linjer) för förändringar i minst Z-värdet över tid </td></tr></table></div> <p>Figure 1. Example of VGA screen layout.</p>	Task_ACC - skriver ut accelerometervärdena, en gång i sekunden X = +128 Y = +128 Z = +000	Task_ACC_filter - skriver ut medelvärdena på 10 sampel X = +096 Y = +130 Z = -002	Task_time - skriver ut tiden från programstart 00:33	Task_plot - ritar graf (linjer) för förändringar i minst Z-värdet över tid 	
Task_ACC - skriver ut accelerometervärdena, en gång i sekunden X = +128 Y = +128 Z = +000	Task_ACC_filter - skriver ut medelvärdena på 10 sampel X = +096 Y = +130 Z = -002					
Task_time - skriver ut tiden från programstart 00:33	Task_plot - ritar graf (linjer) för förändringar i minst Z-värdet över tid 					
3	In the console window, information should be printed (logged) when each task is running with the following content: task name, priority, start time, and relevant data.					



4	Protect shared resources (global variables, printf, etc.) with semaphores to get a robust system. Minimize protected areas to avoid <i>priority inversion</i> .	
5	Idle-task should now only have an infinite loop. Do not print the dot from that task.	
6	The task should be scheduled via Sierra RTOS and each task should have a unique priority.	
7	The bag should have "deadline control" on. If the deadline misses occur, optimize the program code and comment on what was changed.	