Alexandre SPRICH - Junyu LIN

TDDD07 Real Time Systems
Lab 1 : Preparatory questions

1) What can happen if the robot is stuck and motors are active ?

If the robot is stuck and motors are active, there is a high risk of the motors being damaged.
This is why there is in the project a task specifically to avoid obstacles : this is the Avoid task, which allows the robot to avoid getting stuck when running into obstacles, by using the \bump" sensor in front of the robot.
But if that happens, we should be ready to help it to get out as fast as possible

2) What do you do if the robot is out of control ?

If the robot is out of control, we have to turn it off by using the power button, or We can also issue a stop command from Raspberry Pi (by using the command **echo -ne "\xAD" > /dev/ttyUSB1**).
But if it doesn't work, we have to pull out the battery, by following the given steps in the compendium.

3) What is the role of the refine task ?

By using the RFID reader, the Refine task allows to refine the robot positioning, using the tags below the wooden surface that have known positions.
There are two sorts of tags : localisation tags (to deduce the location of the robot) and the victim tags (which represents the victims, which the location is unknown).
For each tag, the task has to query the RFID lecture and check whether the tag is a localisation tag or not.
If the tag is recorded in the localisation database, then the particle lter is updated with this new information.
If a tag is found which is not a localisation tag, then this represents a victim being found, so this information is forwarded to the Report task.

4) How often does the Avoid task need to run ?

According to the temporal system requirements, the Avoid task must be run at least every 100-150ms (to avoid overheating the motors).

5) Can tasks in the robot agent be preempted ?

Like the robot agent code runs as a single Raspberry Pi process, the task scheduler in the robot agent has no possibility to preempt currently running tasks.

6) **In which file(s) will solutions for lab 1 go ?**

For this 1st lab, we'll have to work on the task scheduling of the robot agent. So the program we'll write will be in the task scheduler, in the file **scheduler.c**

7) **What is a metric ?**

A metric is a measurable parameter like time, speed, length, frequency and so on...

8) **What metrics do you plan to use to show your WCET estimates are good enough ?**

We'll test our WCET estimates by measuring the time and the frequency of each task.

should measure execution time under different situations like putting the robot above the victim tag or at a corner, and add some margin to the worst case.

9) **How do you plan to evaluate and measure the performance of the system as defined in Appendix F ?**

We build up an initial schedule to measure execution time of each task. Based on the execution time data, we reschedule to fulfil the timing requirements. Combining a timer with each task (using our built function 'combined_task'), we measure the execution time of each task according to our new schedule. The test runs through the whole schedule for 20 iterations over different locations like near a wall, at a corner or above a victim tag. Thus, the average execution time are taken and the maximum values are recorded as WCET(defined in the beginning of the scheduler.C). Then we use WCET data to refine our schedule. Then we did the execution time measurement again and checked the deadline overruns (within the function 'combined_task') and accuracy of finding victim tags(with the function 'victim_distance_mea').

The final schedule is attached as a separated pdf file.