Embedded Systems (COM3039-N)

ICA Component 1 2017-18

From the module guide

"Component one (70%) involves the individual completion of a case study in which elements of an embedded system design must be implemented in a programming language with real-time support. This component assesses learning outcomes 1, 2, 5 and 7, and includes some timing analysis for real-time applications:

- The completion of a supplied case study design.
- An analysis of the real-time requirements of the system.
- Coding of software components in a real-time programming language."

The task

You are to implement a software controller for the traffic lights controlling a cross-roads. You will be provided with a simulation of the hardware involved for both Windows and Linux.

There are four sets of traffic lights, labelled *north, south, east* and *west*.

Each set comprises;

- red, amber and green lights for motor vehicles,
- red and green lights for pedestrians,
- a button for pedestrians to press,
- a "wait" light for pedestrians,
- a "priority" sensor for emergency vehicles.

You will be provided with;

- *hwif.ads*, an Ada package specification that provides memory-mapped input and output for all the components described above,
- a library file to link against your project which implements the simulated hardware interface,
- the simulation itself.

Your source files should be placed in the icasrc subdirectory. You will need to run the simulation alongside your software. The memory-mapping is implemented via a shared file in the current directory.

The starting state of the system is all red traffic and red pedestrian lights are lit and all others are unlit.

Safety requirements

Your software must ensure that at all times:

1. all pedestrian red lights are lit unless all traffic lights are red;
2. at most one of each pair of red and green pedestrian lights may be lit;

3. at most one of each pair of green pedestrian and wait lights may be lit;
4. each set of traffic lights should illuminate in exactly the conventional (UK) order: red, red-amber, green, amber, then repeat - no other patterns should ever be lit;
5. the amber phase for each traffic light must be at least 3 seconds;
6. if the north or south green traffic light is lit, then the east and west red traffic lights must be lit and no other east or west traffic lights may be lit;
7. if the east or west green traffic light is lit, then the north and south red traffic lights must be lit and no other north or south traffic lights may be lit;
8. the traffic green phase must be at least 5 seconds;
9. the pedestrian green phase must be at least 6 seconds; and
10. a traffic green phase for the direction indicated by a priority sensor must continue for at least 10 seconds beyond when the sensor stops indicating an emergency vehicle.

Timing (liveness) requirements

Your software must ensure that when the system is in the running mode:

11. the delay from a pedestrian pressing a button until receiving a green light should be no more than 60 seconds;
12. the delay from a pedestrian pressing a button until the wait light is lit should be no more than 0.2 seconds;
13. the delay between a traffic green light turning off until it is next lit must not be more than 60 seconds; and
14. the delay between a priority sensor first indicating the presence of an emergency vehicle until that green traffic light is lit should be no more than 10 seconds.

Additional requirements

Your software must ensure that it:

15. does not modify the emergency vehicle sensor inputs;
16. does not modify the pedestrian button inputs;
17. for each traffic light set, at least one light should be lit; and
18. for each pedestrian light set, at least one light should be lit.

Task limitations

Your software **will be assessed on a Linux lab machine**. To simulate the limited memory and CPU resources

- the CPU time will be recorded by the simulator – the lower the sum of user time and system time, the better;
- memory resources will be limited thus:

  - maximum virtual address space is set to 4,194,304 bytes;
  - maximum data segment size (initialized data, uninitialized data, and heap) is set to 131,072 bytes;
  - maximum process stack size is set to 65,536 bytes;
  - Ada task stack size is set to 16kB; and
  - Ada secondary stack size is set to 16kB; and
- the overall size of your executable (on the Linux lab machine).

Warning: *the Windows version provided to you* **does not** *enforce (all) these limits. You* **should** *test your software on a Linux lab machine* **using the `--controller` option and the `controller_linux` GPR file** *or your mark may be disappointingly low. See the section titled "Notices/reminders/hints" below.*

Bugs and software updates

In the event that bugs are discovered in the software I provide, I will fix them (time permitting) and provide updated versions via Blackboard.

Make sure that you are using the current release.

Deliverable and deadline

You should submit the following items via Blackboard by the deadline of 2359hrs 25 January 2017:

- in at most one page of A4, submitted as a Word document or PDF, explain how you manage the requirements in "Timing (liveness) requirements" above; and
- the **source files** individually (not as a ZIP file or any other type of archive) from icasrc (only)

in a **single** Blackboard submission.

Do not submit any files provided by me as part of this ICA, any executables or any intermediate compilation outputs.

If you submit multiple attempts, then I will mark the last submission received before the deadline, or if no submissions are received before the deadline, the submission received first.

Advice and assistance

Consult the module tutors during a scheduled session or contact via email.

Notices/reminders/hints

- The simulator (`hwsim`) has a number of options. When running, you can trigger events. Type "`hwsim --help`" for details.
- When opening a project in GPS, I strongly recommend that you open the project file (.gpr) file. Otherwise, it seems to get quite confused sometimes.
- When opening **this project**, open `controller_windows.gpr` or `controller_linux.gpr` (depending on your operating system).

- You do not need to use GPS. You can directly use the command-line, e.g.,

  ```
  gprbuild -P (gpr filename)
  ```

- Make sure that the simulator and your controller have the same current directory when they run. (Otherwise, they will use different shared memory files….)
- For testing (on Linux), I will use the "`--controller`" option of the simulator. This starts your controller program at a suitable moment and **enforces (some of) the memory limits**. So you should test your code using this option: Example:

  ```
  ./hwsim --time 1800 --controller controller ctrl-output --random
  ```

  This assumes that your controller is called "`controller`" and is in the same directory. Any output generated from your controller is saved in "`ctrl-output`".

- When assessing your controller, I will run it and the simulator for 30 minutes (approximately 72,000 "frames").
- I will use the same sequence of events for all students.

Assessment criteria

This ICA accounts for 70% of the overall mark for the module. The following marking scheme will apply to this ICA (scored out of 100). The criteria below is necessarily incomplete, as we cannot anticipate every possible ICA submission.

[TST] - the results of automated testing of the requirements.

[CPU] - the amount of CPU time reported by my test harness.

[EXE] - the size of the executable.

[TMG] - your answer regarding "Timing (liveness) requirements".

Marks from and values in the following tables will be adjusted up or down if it would be significantly unfair not to do so in the opinion of the academic staff.

TST criteria (/60)

A++ 95% 57	All scores are 0.
A+ 85% 51	Scores 11, 12, 13 and 14 are <= 150. All other scores are 0.
A 75% 45	Scores 11, 12, 13 and 14 are <= 400. All other scores are 0.
B 65% 39	Scores 11, 12, 13 and 14 are <= 1500. All other scores are 0.
C 55% 33	Scores 1, 4, 6 and 7 are 0. Scores 2 and 3 are <= 100. Scores 5 and 9 are <= 5. All other scores are <= 50,000.
D 45% 27	Scores 1, 4, 6 and 7 are <= 100. Scores 2 and 3 are <= 1000. Scores 5 and 9 are <= 50. All other scores are <= 50,000.
E 35% 21	Scores 1, 4, 6 and 7 are <= 1000. All other scores are <= 60,000.
F 0%-20% 0-12	None of the above.

CPU criteria (/10)

If your grade for the TST criteria is E or lower, then this mark is 0.

If you grade for the TST criteria is D or higher, then:

< 0.1	10
[0.1, 0.25]	5
>= 0.25	0

EXE criteria (/10)

If your grade for the TST criteria is E or lower, then this mark is 0.

If you grade for the TST criteria is D or higher, then:

< 1,430,000	10
[1,430,000, 1,900,000]	5
>= 1,900,000	0

TMG criteria (/20)

A 75%/85%/95% 15/17/19	*Excellent* A very clear and readable report which gives an excellent answer with clear reasoning.
B 65% 13	*Substantially correct/appropriate (based on taught material & module requirements)* A clear and readable report that gives a sensible, reasoned answer.
C 55% 11	*Minor errors/omissions/issues* A clear and readable report that gives a mostly sensible answer, or a sensible answer with limited reasoning.
D 45% 9	*Major errors/omissions/issues* A report that is difficult to read/comprehend or gives at best a limited answer.
E 35% 7	*Unsatisfactory* A report that is difficult to read/comprehend and gives a very limited answer.
F 0%-20% 0-4	*Inadequate* A report that is very difficult to read and comprehend, and makes no real attempt at answering the question.