Assignment for Credit

Course Code : ENG 5220	Course Name : *Real Time Embedded Programming*
Type : Technical Report *Oral Presentation &* *Social media campaign*	Title of Assignment : Development, design, construction and promotion of a product requiring realtime operation
% of final course mark : *100%*	Lecturer : *Bernd Porr & Nicholas Bailey*

Marking

1. 20% will be given to the initial pitch of the project (every team has 5 mins and two slides) and initial github pages up and running (special timeslot, see moodle). We assess here the originality/usefulness of the work, if it has a solid realtime requirement and the quality of the presentation.
2. 20% of the final submitted work will be based on the quality of the hardware design and its reliability. Is this the right choice of components? In particular is the design contemporary by using carefully selected currently produced sensors rather than just what's "in stores"? Has analogue design been minimised and moved to digital processing? Is the design robust? Has it sufficient protection? Is the safety of the user guaranteed?
3. 20% of the final submitted work will assess the realtime responsiveness of the software and how this has been achieved which includes if processing of events has been achieved by waking up threads and in general employing event driven code using timers, signals, threads and/or kernel space interrupt driven coding.
4. 20% of the final submitted work will be based on the use of formal design methods, revision control, committing, branching, creating releases , testing and project planning. Clear division of labour and documentation of the work. Formal methods have been used for use-case, high- and low-level software and test-vector generation. Has git been used as a revision control system or just to ``upload'' code? Has git been used to do revisions, track bugs and has there been a release strategy?Has the issue tracker system been used? Have unit tests been used? Is the design formally documented?
5. 20% of the marks are devoted to the promotion of the work: has the project been properly presented on github so that it catches the eye of a user? Is the hard/software described in a way that other people can reproduce it? Has the project been advertised on social media and has it been picked up by online publications such as hackaday? Has a social media account been created and has it created a buzz around it? Has the project a license?

All items above will marked on the 22 point scale, according to the performance indicators written overleaf. Consideration will be given to the inclusion of Aims and Objectives and clarity of presentation.

Submission & Return

The submission is online via moodle where you submit the link to the github page which contains your report, code, hardware documentation and links to social media. In addition every team should be submit a single page just containing the names of the team members and a link to moodle to the School of Engineering Learning & Teaching Office in room 621 of the James Watt South building with a coursework form (http://userweb.eng.gla.ac.uk/moodle/) to confirm that the submission is all your

own work.

Make sure that each group member's area of responsibility is clearly marked.

Note that University policy on late submission of work without good cause is that the grade will be reduced by two secondary bands (e.g. from 'B1' to 'B3 or 'A5' to 'B2') for each working day, or part of a working day, after the submission deadline. Work submitted more than five days after the deadline will receive an 'H' grade. If you are unable to submit work on time due to good cause, you should contact me as soon as is possible to seek a deferral.

Submission deadline : 15 April 2019

Results & Feedback

Feedback & results about the initial pitch will be available one week after the presentation and will come from both lecturers. You will receive a list of 'common areas of improvement' about your proposed project. This will give you feedback on where you did well, and where you could improve.

You can collect feedback about your final work from the School of Engineering Learning and Teaching office (JWS 621). This feedback will be structured according the 4 marking criteria above covering the final work and will comment on every section.

Results will be posted (anonymously by registration number) on Moodle after feedback has been deposited for collection from the L&T office.

1. Presentation

Grade range	**A1, A2**	**A3, A4, A5**	**B1, B2, B3**	**C1, C2, C3**	**D1, D2, D3**	**E1, E2, E3**	**F, G, H**
Aggregation	**22, 21**	**20–18**	**17–15**	**14–12**	**11–9**	**8–6**	**5–0**
Score							**(maybe CR)**
Delivery	Could present at a conference with no further training	Confident delivery, clear speech, no hesitation, held attention	Good delivery, only minor flaws/hesitation	Significant lapses in delivery but satisfactory overall	Hard to follow significant parts of the talk	Couldn't make out anything without difficulty	Impossible to learn anything
Slides	Of professional conference quality	Excellent slides, attractive appearance, information well presented	Good slides, only minor flaws such as poor layout or plots with illegible axes	Some slides had illegible text or incomprehensible illustrations	Poor slides, hard to read or deduce content	No effort made to prepare appropriate slides	No slides (consider CR)
Originality	A novel product idea with clear market appeal	Impressive idea which is genuinely novel	Idea appropriate to the brief	Indea generally satisfactory but not clear what is original here	Idea not clear and hard to judge	Generally inadequate or incorrect content	No worthwhile idea(consider CR)
Realtime	Professional, quantitative realtime assessment	Clear case for realtime processing	Satisfactory case for realtime processing. Mostly qualitative.	Realtime demands not completely clear.	Poor case for realtime procesing, lacking major aspects	Minimal understanding of realtime processing.	No understanding of realtime processing.
Response to questions	Supervisor learnt from response to questions	Confident and informed response to all questions	Good response to questions but occasionally unconvincing	Satisfactory response to most questions	Had difficulty answering most questions	Required prompting for any answer	Unable to answer any questions satisfactorily

2. Hardware

Grade range	**A1, A2**	**A3, A4, A5**	**B1, B2, B3**	**C1, C2, C3**	**D1, D2, D3**	**E1, E2, E3**	**F, G, H**
Aggregation	**22, 21**	**20–18**	**17–15**	**14–12**	**11–9**	**8–6**	**5–0**
Score							**(maybe CR)**
Reliability/quailty of the hardware design	Industry standard of a prototype: PCB desgn, protection, connectors, mechanical desin.	Perfectly working prototype but might fail in the long run or under stress tests.	Solid technical desin but with minor flaws in design.	Competent technical skills but device would fail in a real world setting.	Satisfactory design but with significant shortcomings in some areas.	Design shows major weaknesses.	Showed few or none of the skills expected of a graduate (consider CR)
Contemporary design choices	Industry standard choices by using up to date (preferably digital) sensors / actuators	Contemporary design choices minimising analogue circuits.	Good design using contemporary robust devices.	Too much analogue circuitry drawing from old designs.	Circuit works but uses outdated circuits	Almost no effort to research in into sensors / actuators	Achieved virtually nothing (consider CR)

3. Realtime responsiveness

Grade range	**A1, A2**	**A3, A4, A5**	**B1, B2, B3**	**C1, C2, C3**	**D1, D2, D3**	**E1, E2, E3**	**F, G, H**
Aggregation	**22, 21**	**20–18**	**17–15**	**14–12**	**11–9**	**8–6**	**5–0**
Score							**(maybe CR)**
Assessment of latencies in the application context	Professional quantitative assessment and tolerances	Good quantitative assessment of the realtime demands	Correct assessment of requirements but smaller shortcomings	Assessment of the latencies partially wrong or not completely considered.	Latencies seriously assessed.	Almost no effort to research in into latencies.	Achieved virtually nothing (consider CR)
Realtime coding	Production level realtime coding using threads/timers/signals and kernel interrupts	Perfectly working prototype but minor shortfalls in structure, doc or reliability.	Solid realtime coding but with smaller coding issues causing small noticable latencies.	Realtime coding has shortcomings in responsiveness, timing and sampling of signals.	Significant shortcomings in the realtime coding resulting in long latencies.	Design shows major weaknesses in realtime processing utilising delays / blocking code..	Showed few or none of the skills expected of a graduate (consider CR)

4. Revision control and project management

Grade range	**A1, A2**	**A3, A4, A5**	**B1, B2, B3**	**C1, C2, C3**	**D1, D2, D3**	**E1, E2, E3**	**F, G, H**
Aggregation	**22, 21**	**20–18**	**17–15**	**14–12**	**11–9**	**8–6**	**5–0**
Score							**(maybe CR)**
Revision control	Professional use revision control with regular commits, branching & merging	Good use of revision control with detailed commits	Use of revision control but shortcomings in commits and development on master	Only work on master without any safeguards and shortcomings in commits	Only few commits on the master branch with generic comments.	Used github only as an upload site with no collaborative effort	Achieved virtually nothing (consider CR)
Project management	Exemplary; could not have done better with the time and resources available	High-quality planning, made excellent use of time and resources available	Good planning and use of resources with only minor deficiencies	Satisfactory planning but could clearly have made better use of resources.	Poor planning and use of resources; did not always follow directions	All over the place; required continual direction from supervisor	Did only what the supervisor told him or her, if that
Software Design	Formal methods used at all stages from use-case to deployment. Extensively tested. Multiple possible system designs critically and comprehensively analysed.	Comprehensive, multimodal analysis of requirements and structure from use-case analysis to structure and timing diagrams. Rigorous test-vector genertion.	Considered overall system structure advises implementation. Appropriate use of formal methods.	Some use of formal design techniques. Little evidence that the design process advises the implementation.	Some incomplete or erroneous design documents; design is documentary rather than creative.	Incomplete documentation of system design. Major omissions or errors in the design documents.	Limited or no evidence of a formal approach to design.
Reliability / Testing / Bug fixing	Professional testing approaches with unit tests, issue tracking, fixing	Good test scenarios which unit tests	Satisfactory testing and debugging but smaller shortcomings	Testing only in some cases but clearly some are left out.	Poor testing just in a qualitative manner,	No explicit testing but just report of success.	Achieved virtually nothing (consider CR)

5. Documentation and PR

Grade range	**A1, A2**	**A3, A4, A5**	**B1, B2, B3**	**C1, C2, C3**	**D1, D2, D3**	**E1, E2, E3**	**F, G, H**
Aggregation	**22, 21**	**20–18**	**17–15**	**14–12**	**11–9**	**8–6**	**5–0**
Score							**(maybe CR)**
Quality of the content	Professional level of documentation comparable to other github professional projects	Comprehensive coverage with no significant omissions	Good coverage with only minor omissions	Covered much of the project but with significant omissions	Major omissions; large parts of project not covered	Only a little material relevant to project	Nothing of substance (consider CR)
Quality of argument	Could stay on github without further work	Arguments well presented; results clear and accessible	Results critically assessed	Discussion of results lacks insight	Discussion of results at only a very low level	Discussion perfunctory	No discussion apparent
Illustrations and video content	Worthy of publication	Well-chosen, illuminating and attractively formatted illustrations and excellent video	Good illustrations that enhance the report and an eye catching video	Illustrations satisfactory but could be drawn or chosen better; too few illustrations. Video could have clearer message.	Poor illustrations or mostly from WWW. Video taken from other websites and/or combined.	Images only from WWW and/or video.	No illustrations (consider CR) No video.
PR / social media strategy / release strategy	Perfectly devised strategy on all channels and targeting the right audience	Well devised strategy covering all relevant channels and target audience.	PR strategy reflects a good amateur project but has shortcomings for a prof product	PR OK for a local group of friends and followers but has shortcomings reaching beyond it	Poor PR just involving a few last minute posts on social media. No clear strategy.	PR strategy just limited to github.	No PR (consider CR)

University of Glasgow | School of Engineering

§1 Task Overview

Aims

Development, design, construction and promotion of a product requiring realtime operation.

Objectives

- Propose a product which requires realltime processing and solves a real world task
- Develop product hardware connecting to the Raspberry PI as proof of concept
- Develop realtime software in C++ (web/mobile apps in scripting languages)
- Create, maintain, schedule and document the project using git version control, tests and quality management
- Promote the final product via github, social media and live demos

§2 Task Requirements

The task is to present an end user product which requires realtime processing. This will be build around the Raspberry under Debian Linux. It needs to be a project which solves a real world problem, for example, watering plants while away on holiday or a mattress which senses if a person sleeps well. This needs to be achieved within a budget of max £75 for Electronic store orders.

In technical terms this means that the raspberry PI needs to measure physical values, plot them on the screen, allow mouse interaction to change parameters and that it generates meaningful outputs. All this in realtime. You need to make your own PCB (no ready made solutions allowed) and software. At the end you should have a standalone embedded application which boots up and performs your chosen task.

The raspberry PI has no A/D converter but it has a general purpose SPI/I2C/I2S/GPIO interface which connects to a large number of sensors. Generally aim to use digital sensors using a digital protocol.

Main coding language must be C++. The operating system is Linux on the Raspberry PI. Code must be written in an object oriented fashion with a testing framework i.e. unit testing. Only web servers and mobile phone apps are permitted to be written in a scripting language (PHP, js, Python, JAVA, swift, ...). Anything else needs to be in C++ or C for kernel programming. The code needs to be event driven -- either in userspace with signals and/or waking up threads and/or interrupt driven in kernel space.

Form groups of three and every person should have distinct roles. On moodle is a wiki where every team enters their names, matric numbers and links to github where their entire project is hosted.

Whilst creative lateral thinking is always welcome in Masters level courses, it is possible to take shortcuts in creating an application which mean that it is no longer realtime, or is otherwise trivial in nature, and thus does not show mastery of the Intended Learning Outcomes of the course. We set out here requirements for the work, which if you ignore will ensure that your project does not fulfil the brief and is liable to receive few if any marks. In particular the following criteria pose a strong risk that the group will receive zero marks:

- program goes into wait state and becomes unresponsive
- using wait statements to establish timing instead of switching threads or load balancing
- trivial work selling just with public relations but no substance
- no indication of version control
- not using C++ as the main coding language (remember: scripting is only allowed for web services and apps)

Original approaches are nonetheless encouraged. You are encouraged to discuss them with the course co-ordinator at an early stage to be sure that they won't be considered off-topic and thus could result in a very low grade. Conversely, you should also discuss with the course coordinator if your project is realistically achievable given the constraints of the course and adjust your goals/objectives accordingly early in the project (i.e. by the pitching session).

§3 Formal contact hours and independent work

You'll spend 33 hours in the lab under supervision. There are also 11 hours of lectures you need to attend. In addition you'll need to work both independently in the lab and do independent study in the remaining 156 hours allocated to this class. This work requires a high degree of independent work while the lab sessions shall be used to get advice, guidance and feedback from both the academics and teaching assistants.

Note that practical work (i.e. soldering, hardware development, etc) needs to be done during office hours 9am – 5pm. Outwith these hours only PC work is allowed such as coding and documentation.