

Assignment for Credit

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

Marking

- 10% will be awarded for 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.
- 25% of the credit for the final submitted work will be based on the way the code is structured, that it is divided up in classes allowing encapsulation of data, using data structures in a failsafe way, receiving data and releasing data in a safe way and in general guaranteeing high reliability and ease of maintenance.
- 25% of the credit for the final submitted work will assess the realtime responsiveness of the software and how this has been achieved. This includes whether 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 in preference to polling or other less suitable methods.
- 25% of the credit for the final submitted work will be based on the use of revision control, committing, branching, creating releases, testing and project planning. Marks will be awarded for demonstrating clear division of labour and documentation of the work. 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?
- 15% 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 potential 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 be 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.

信贷转让

课程编号:ENG5220	课程名称：实时嵌入式编程
类型：技术报告 口头陈述& 公共关系组	任务名称：需要实时操作的产品的开发、设计、建造和推广
最终课程分数%:100%	讲师：BerndPorr&NickBailey

Marking

- 10%将被授予项目的初始间距（每个团队有5分钟和两张幻灯片）和初始github页面启动并运行（特殊时间段，请参阅moodle）。我们在这里评估工作的创意有用性，如果它有一个坚实的实时要求和演示的质量。
- 最终提交工作的25%的信用将基于代码的结构方式，它被划分为允许封装数据的类，以故障安全的方式使用数据结构，以安全的方式接收数据和释放数据，
- 最终提交作品的25%的信用将评估软件的实时响应能力以及如何实现这一目标。这包括事件的处理是否已经通过唤醒线程来实现，并且通常使用使用定时器，信号，线程或内核空间中断驱动编码的事件驱动代码，而不是轮询或其他不
- 最终提交工作的25%信用将基于修订控制，提交，分支，创建发布，测试和项目计划的使用。将颁发标志，以证明明确的分工和工作文件。Git是否被用作修订控制系统或只是为了`上传"代码？Git是否被用来做修订，跟踪错误，是否有发布策略？是否使用了问题跟踪系统？是否使用了单元测试？
- 15%的标记用于推广工作：该项目是否已在github上正确呈现，以便吸引潜在用户的眼球？硬软件是否以其他人可以复制的方式描述？该项目是否在社交媒体上做过广告，是否被hackaday等在线出版物所采用？是否创建了一个社交媒体帐户，并在其周围产生了嗡嗡声？这个项目有许可证吗？

以上所有项目将根据背上写的性能指标在22点刻度上标记。将考虑列入目标和目标以及说明的明确性。

Submission & Return

The submission is online via moodle where you submit the link to the github page which contains your report, code, hardware 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 their github 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 : 17 April 2023

提交及申报

提交通过moodle在线，您将链接提交到github页面，其中包含您的报告，代码，硬件和社交媒体链接。此外，每个团队都应该提交一个仅包含团队成员姓名的页面以及指向其github的链接，以确认提交的所有内容都是您自己的工作。

确保每个小组成员的责任范围都有明确的标记。

请注意，大学对无正当理由逾期提交作品的政策是，在提交截止日期后，每个工作日或一个工作日的一部分，成绩将减少两个二级等级（例如从'B1'到'B3或'A5'到'B2'）。在截止日期后五天以上提交的作品将获得"H"级。如果你因正当理由不能按时提交工作，你应该尽快与我联系，寻求延期。

截止报名日期：2023年4月17日

Results & Feedback

Feedback & results about the initial pitch will be available after the presentation and will come from both lecturers.

You will receive feedback about your final work via email. This feedback will be structured according the 4 marking criteria above covering the final work and will comment on every section.

结果与反馈

关于初始音高的反馈和结果将在演讲后提供，并将来自两位讲师。

您将通过电子邮件收到有关您最终工作的反馈。此反馈将根据上述4个评分标准进行结构化，涵盖最终工作，并将对每个部分发表评论。

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 Score	22, 21	20–18	17–15	14–12	11–9	8–6	5–0 (maybe CR)
Delivery	Could present at a conference with no further training	Confident delivery, only minor hesitation, held attention	Good delivery, such as 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 as poor layout or plots with illegible axes	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	Idea generally satisfactory but not hard to judge	Idea not clear and difficult to judge	Generally inadequate or clear what is original here	No worthwhile idea(consider CR)
Realtime	Professional, quantitative realtime assessment	Clear case for realtime processing	Satisfactory case for realtime processing. Mostly qualitative.	Poor case for realtime processing, not completely clear.	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 satisfactorily

1. Presentation

职系范围	A1, A2	A3, A4, A5	B1, B2, B3	C1, C2, C3	D1, D2, D3	E1, E2, E3	F, G, H
Score	22, 21	20–18	17–15	14–12	11–9	8–6	5–0 (maybe CR)
Delivery	可以在没有进一步培训的情况下出席会议	自信的表达, 清晰的演讲, 毫不犹豫, 引起了人们的注意	良好的交付, 只有轻微的缺陷, 失误, 但整体令人满意	交货方面有重大失误, 但整体令重要部分人满意	很难理解谈话的内容	什么都看不懂	不可能学到任何东西
Slides	专业会议质素	优秀的幻灯片, 有吸引力的外观, 信息很好地呈现	良好的幻灯片, 只有轻微的缺陷, 如布局不良或坐标轴难以辨认	有些幻灯片有难以辨认的文字或插图	糟糕的幻灯片, 难以阅读或推断内容	没有努力准备适当的幻灯片	没有幻灯片 (考虑CR)
Originality	具有明确市场吸引力的新颖产品理念	令人印象深刻的想法, 这是真正新颖的	适合简报的想法	印地亚总体上令人满意, 但不清楚这里的原创内容	思路不清晰, 难以判断	一般不充分或不正确的内容	没有值得的想法 (考虑CR)
Realtime	专业、定量的实时评估	实时处理的明确案例	满意的情况下进行实时处理。大多是定性的。	实时要求不完全清楚。	实时处理的情况很差, 缺乏主要方面	对实时处理的最小了解。	不了解实时处理。
回答问题	主管从回答问题中学习	对所有问题作出自信和知情的回答	对问题的回答很好, 但偶尔不具说服力	对大多数问题的满意的答复	回答大多数问题都有困难	需要提示任何答案	无法满意地回答任何问题

2. Structure of the code

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 (maybe CR)
Score							
Optimal choice of classes (SOLID)	Classes have clear responsibilities, interfaces are segregated to be client specific, dependency inversion, obey the Liskov Substitution Principle. Minor issues but still professional documented in an intuitive way.	Classes have clear responsibilities, interfaces are segregated to be client specific, dependency inversion, obey the Liskov Substitution Principle. Minor issues but still professional production standard.	Generally following the SOLID principles but either one is violated or documentation does not demonstrate that they have been taken into consideration.	Some SOLID principles haven't been applied and/or there are violations of the principle. Documentation has flaws which makes it hard to see if/how they have been applied.	Serious flaws in the implementation of SOLID and most principles haven't been applied. There is little mention in the documentation about the class choices.	Not clear whether SOLID or little to has been applied mark at all.	No application of SOLID or little to has been applied mark at all.
Encapsulation of data in classes and safe use of getters and setters and data management.	Clear public interfaces are defined, the data is private and getters and setters provide a safe interface to the client. However, some minor flaws for example in terms of safety, timing and choice of internal data structures.	Public interfaces are defined, the data is private and getters and setters provide a safe interface to the client. However, some minor flaws for example in terms of safety, timing and choice of internal data structures.	Generally data is encapsulated and the internal storage of data is appropriate but there smaller issues with the getters / setters, not checking for fault conditions or the internal data storage could be more efficient.	Significant problems with encapsulation such as public variables and no fault checking. Data storage/management is inefficient.	Serious flaws in encapsulation with public variables being accessed, no clear getter/setter interfaces and data is stored in not appropriate structures.	No encapsulation in use of global variables or class-but classes work less coding.	No classes used, by accessing variables and calling member functions. No use of public / private variables & members.
Failsafe memory management	Memory management is completely leak free.	Memory management is leak free but uses new/delete where it could be avoided.	Excessive use of new/delete where new/delete could be used.	Clearly there is a lack of care of C++ instances and tidy up memory copy constructors could be used.	Serious flaws of memory management with eventual crash.	Serious flaws in memory management leading to out of memory.	No memory management at all or nothing to mark. allocations.

2.代码的结构

职系范围	A1, A2 22, 21	A3, A4, A5 20–18	B1, B2, B3 17–15	C1, C2, C3 14–12	D1, D2, D3 11–9	E1, E2, E3 8–6	F, G, H 5–0
类的最佳选择(固体)	类有明确的责任，接口被隔离为特定于客户端，依赖倒置，服从Liskov替换	类有明确的责任，接口被隔离为特定于客户端，依赖倒置，服从Liskov替换	通常遵循坚实的原则，但其中一项被违反或文件没有证明它们已被考虑在内。	一些坚实的原则没有得到应用，或者有违反该原则的行为。文档有缺陷，这使得很难看到它们是如何应用的。	实和大多数原则的实施方面的严重缺陷没有得到应用。文档中几乎没有提到类的选择。	不清楚是否已应用固体。有些方面似乎适用，但没有直接的证据或文件表明这一点。	没有应用固体或很少标记。
在类中封装数据，安全使用getter和setter以及数据管理。	原则。小问题，但仍专业生产标准。	定义了公共接口，数据是私有的，get和setter为客户端提供了一个安全的接口。内部数据结构是有效的get和setter提供快速的accesscompuation。	通常，数据被封装，数据的内部存储是适当的，但gettersetter的问题较小，不检查故障条件或内部数据存储可能更有效率。	封装的重大问题，如公共变量和没有故障检查。数据存储管理效率低下。	在访问公共变量时的封装中存在严重缺陷，没有明是类通过访问getter和setter接口和数据成员函数来工作。	在使用的类中没有封装，但是类通过访问变量和调用成员函数来工作。不使用公共私有变量和成员。	没有使用类，使用全局变量或无类编码。
故障安全内存管理	内存管理完全无泄漏。	内存管理是无泄漏的，但在可以避免的地方使用新的删除。	过度使用可以使内存分配的照顾构造函数的newd。elete。	显然，缺乏整理用C++实例和复制内存分配的照顾构造函数的newd。elete。	内存管理的严重缺陷与最终崩溃导致。	内存管理中的严重缺陷导致内存不足。	根本没有内存管理或没有标记。

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 (maybe CR)
Score							
Assessment of latencies in the application context and appropriate design decisions	Professional quantitative assessment and tolerances leading to clear coding decisions	Good quantitative assessment of the realtime demands leading to good coding decisions with small omissions.	Correct assessment of requirements but smaller shortcomings and resulting smaller issues in terms of coding decisions.	Assessment of the latencies partially wrong or not completely justified of the realtime coding proposed coding strategy.	Latencies not seriously assessed and thus no justification of the realtime coding framework is not well thought through.	Almost no effort to research into latencies and their knock on effect on coding.	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 minor shortfalls in smaller coding issues causing small noticeable 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)

3.实时响应能力

职系范围	A1, A2	A3, A4, A5	B1, B2, B3	C1, C2, C3	D1, D2, D3	E1, E2, E3	F, G, H
Score	22, 21	20–18	17–15	14–12	11–9	8–6	5–0 (maybe CR)
评估应用环境中的延迟和适当的设计决策	专业的定量评估和公差，导致清晰的编码决策	对实时需求进行良好的定量评估，从而在很小的遗漏下做出良好的编码决策。	要求的正确评估，但较小的缺点，并在编码决策方面产生较小的问题。	对延迟的评估部分错误或不完全考虑，并且建议的编码框架没有得到很好的考虑。	延迟没有认真评估，因此没有理由的实时编码策略。	几乎没有努力研究延迟及其对编码的影响。	几乎一无所获（考虑CR）
实时编码	使用线程计时器签署als和内核中断的生产级实时编码	完美的工作原型但在结构、定时性和采样性方面有轻微的不足。	可靠的实时编码，但较小的编码问题导致小的明显延迟。	实时编码在信号的响应性、定时性和采样性方面存在不足。	实时编码的显著缺点导致长延迟。	设计显示了利用延迟阻塞代码的实时处理的主要弱点。	表现出很少或没有毕业生所期望的技能（考虑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 (maybe CR)
Score							
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 safeguards and commits and development on master	Only work on master without any safeguards and shortcomings in generic comments.	Only few commits on the master branch with no 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
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)

4. 修订控制和项目管理

职系范围	A1, A2	A3, A4, A5	B1, B2, B3	C1, C2, C3	D1, D2, D3	E1, E2, E3	F, G, H
Score	22, 21	20–18	17–15	14–12	11–9	8–6	5–0 (maybe CR)
修订控制	专业使用定期提交、分支和合并的修订控制	使用详细提交的修订控制	修订控制的使用，但在提交和开发上的缺点	只在master上工作，没有任何保障和提交中的缺点	只有很少的提交在主分支与通用注释。	Github仅用作上传网站，无需协作	几乎一无所获（考虑CR）
项目管理	示范性的；用可用的时间和资源做得再好不过了	高质量的规划，充分利用可用的时间和资源	良好的资源规划和使用，只有轻微的不足之处	令人满意的规划，但显然可以更好地利用资源。	资源的规划和使用不力；不总是遵循指示	到处都是，需要主管的持续指导	只做了主管告诉他或她的事情，如果
测试错误修复	具有单元测试、问题跟踪、修复的专业测试方法	单元测试的良好测试场景	满意的测试和调试，但较小的缺点	仅在某些情况下进行测试，但显然有些被排除在外。	质量差的测试	没有明确的测试，只是成功的报告。	几乎一无所获（考虑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 Score	22, 21	20–18	17–15	14–12	11–9	8–6	5–0 (maybe CR)
Quality of the content	Professional level of documentation comparable to other github prof 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 to project	Only a little material relevant (consider CR)	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.	Poor illustrations or mostly from WWW. Video film has a poor quality.	Images only from WWW. The video has low quality in terms of narrative and presentation.	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	PR OK for a local group of friends and followers but has shortcomings on social media.	Poor PR just involving a few last minute posts reaching beyond it	PR strategy just limited to github.	No PR (consider CR). No clear strategy. prof product

5.文件及公关

职系范围	A1, A2	A3, A4, A5	B1, B2, B3	C1, C2, C3	D1, D2, D3	E1, E2, E3	F, G, H
Score	22, 21	20–18	17–15	14–12	11–9	8–6	5–0 (maybe CR)
内容的质量	与其他github项目相当的专业文档水平	全面保障，无重大遗漏	良好的复盖范围，只有轻微的遗漏	涵盖了大部分项目，但有重大遗漏	重大遗漏；项目的大部分未包括在内	只有一点与项目相关的材料	没有实质内容（考虑CR）
论证的质量	可以留在github上而无需进一步的工作	论证良好，结果清晰易懂	批判性评估结果	对结果的讨论缺乏洞察力	只在很低的水平上讨论结果	讨论敷衍	没有明显的讨论
插图和视频内容	值得出版	精心挑选，具有启发性和吸引力的格式插图和优秀的视频	良好的插图，增強了报告和引人注目的视频	插图令人满意，但可以更好地绘制或选择；插图太少。视频可以有更清晰的信息。	可怜的插图或主要来自WWW。视频电影在叙事和呈现方面质量很差。	仅来自WWW的图像。视频质量很差。	没有插图（考虑CR）没有视频。
公关社交媒体策略发布策略	在所有渠道上完美设计策略，并针对正确的受众	精心设计的策略涵盖所有相关渠道和目标受众。	公关策略反映了一个很好的业余项目，但对于prof产品有缺点	对于一群当地的朋友和追随者来说，公关是可以的，但也有超出它的缺点	可怜的公关只是涉及社交媒体上的最后一分钟帖子。没有明确的策略。	PR策略仅限于github。	No PR (consider CR)

§1 Task Overview

Aims

Development and promotion of a product requiring realtime operation.

Objectives

- Propose a product which requires realtime processing and solves a real world task
- Select hardware connecting to a Raspberry PI as proof of concept
- Develop realtime software in C++ (web-pages/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 Linux on a Raspberry PI. 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.

In technical terms this means that the Linux system 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. At the end you should have a standalone embedded application which boots up and performs your chosen task.

Your task is to use data acquisition hardware, for example the sound card or on the Raspberry PI sensor boards and digital sensors.

Main coding language must be C++. The operating system must be Linux. Code must be written in an object oriented fashion with a testing framework i.e. unit testing. Only web pages and mobile phone apps are permitted to be written in a scripting language (PHP, js, Python, JAVA, swift, ...).

The code must be event driven -- either in userspace with signals and/or waking up threads and/or interrupt driven in kernel space.

Form groups of four 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

§1任务概述

Aims

开发和推广需要实时操作的产品。

Objectives

- 提出一个需要实时处理并解决现实任务的产品
- 选择连接到树莓PI的硬件作为概念证明
- 用c++开发实时软件（脚本语言的网页移动应用程序）
- 使用git版本控制、测试和质量管理创建、维护、计划和记录项目
- 通过github、社交媒体和现场演示推广最终产品

§2任务要求

我们的任务是展示一个需要实时处理的最终用户产品。这将在树莓PI上围绕Linux构建。它需要是一个解决现实世界问题的项目，例如，在度假时给植物浇水或感觉一个人是否睡得很好的床垫。

在技术术语中，这意味着Linux系统需要测量物理值，在屏幕上绘制它们，允许鼠标交互更改参数，并生成有意义的输出。所有这些都是实时的。最后，您应该有一个独立的嵌入式应用程序启动并执行您选择的任务。

您的任务是使用数据采集硬件，例如声卡或树莓PI传感器板和数字传感器。

主要编码语言必须是C++。操作系统必须是Linux。代码必须以面向对象的方式编写，并具有测试框架，即单元测试。只允许用脚本语言（PHP、js、Python、JAVA、swift）编写网页和手机应用程序。..)。

代码必须是事件驱动的-无论是在用户空间与信号和或唤醒线程和或中断驱动在内核空间。

四人一组，每个人都应该有不同的角色。在moodle上是一个wiki，每个团队都输入他们的名字，入学编号和指向github的链接，他们的整个项目都托管在那里。

虽然创造性的横向思维在硕士课程中总是受欢迎的，但在创建应用程序时可以走捷径，这意味着它不再是实时的，或者在本质上是微不足道的，因此不能显示出对课程预期学习结果的掌握。我们在这里列出了这项工作的要求，如果你忽略了这项工作，就会

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
- not using callbacks to process events
- 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)

Do not hesitate to discuss with the course co-ordinator any original approaches to the assignment you are worried might be off-topic and thus could attract a very low grade.

§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.

§3 Hardware purchases

The budget is £45 per team for orders via the electronics store and/or technician.

确保你的项目没有完成概要，并且很少收到任何标记。特别是，以下标准构成了集团获得零分的巨大风险：

- 程序进入等待状态，变得无响应
- 使用wait语句建立定时，而不是切换线程或负载平衡
- 不使用回调来处理事件
- 琐碎的工作销售只是与公共关系，但没有实质内容
- 没有版本控制的指示
- 不使用C++作为主要编码语言（记住脚本只允许用于web服务和应用程序）

不要犹豫，与课程协调员讨论任何原始的方法，你担心的任务可能是偏离主题，因此可能会吸引一个非常低的成绩。

§3 正式联络时间及独立工作

你会在实验室里呆上33个小时。您还需要参加11小时的讲座。此外，您需要在实验室独立工作，并在分配给本课程的剩余156小时内进行独立学习。这项工作需要高度独立的工作，而实验室的工作则会从学者和助教那里得到建议、指导和反馈。

§3硬件购买通过电子商店或技术人员订购的预算为每个团队45英镑。