

# ENCE260 Embedded Programming Assignment

**Due:** 5.00 p.m. Friday 17<sup>th</sup> October.

**Worth:** 10% of final grade.

**Groups:** Students work in groups of two.

**Purpose:** To have some fun and learn about simple, modular C programming on a small-scale embedded system.



## 1. Introduction

You are to write a small set of programs that will implement a simple multi-user game<sup>1</sup> for your UCFK4 microcontroller boards.

The goals of the assignment are:

1. To program a small-scale embedded system in a modular fashion.
2. To use the modules provided to abstract hardware dependence.
3. To implement a simple real-time infrared communications protocol.
4. To implement at least one C module, each.
5. To follow a coding style standard.

## 6. General instructions

Here's what you need to do:

1. You should have already formed a group using Learn's booking system. If not, you have been allocated a project partner and have been emailed details. Each group member will need to login to <https://eng-git.canterbury.ac.nz/steve.weddell/ence260-ucfk4> at least once so that we can create a git repository you<sup>2</sup>. Your code cannot be marked until you have a git repository and have committed your code to this your repository.
2. Develop a multi-user interactive game for the UCFK4 using the navswitch, display, and IR communications.
3. Demonstrate your program on Friday 16 October between 11:00 and 13:00, i.e. the last day of term.
4. Perform a final commit of your program source code to your git repository by 17:00 Friday 16 October for marking.

## 5. Program submission

1. NB, your game must consist of a single application. If you require different functionality on each board, then this must be determined at run-time, say by pushing a button.
2. Ensure each of your source modules have both group member's names and usercodes in a banner at the start. Your code will be assessed in terms of documentation.
3. Ensure you have followed the required programming style for this project.
4. Ensure you have not changed any of the modules staff have provided.
5. Place all your final application modules in the provided directory: `ucfk4labs/assignment/final/`. Only your modules in this directory will be built, tested, and marked.
6. Ensure that you have a working Makefile that will build your program. Note that a template is provided. Also note that marks will be lost if I cannot build your application by running `make`.
7. Commit your source modules to your subversion repository by 5.00 pm Friday 17<sup>th</sup> October.

---

<sup>1</sup> To help us better assess your work, one of two games is recommended, i.e., either: 1) paper, scissors, rock, or 2) pong.

<sup>2</sup> Don't use the url on the ecewiki UCFK4 software page!

## 8. Assessment

Your programs will be compiled, checked for compiler warnings and errors, and checked for plagiarism. I will award marks based on the following:

1. The modularity of your programs. I expect you to write at least one C module (both the implementation and header module).
2. The readability of your programs (consistent formatting, commenting/documentation, avoidance of embedded constants <magic numbers>, consistent naming, etc.).
3. How well your programs work during a short inspection (from 11:00 on Friday 16 October).
4. How simple your program is to understand. I prefer a simple well structured, well formatted, easy to read program.

## 5. Plagiarism

Every year I detect plagiarism in this assignment. If I detect any hint of plagiarism I will award you zero marks. You may use other code (such as the code I provide) provided you acknowledge who wrote it. You may not use other student's code.

## 6. Suggestions

Programming is fun but not if you leave it to the last minute. Indeed, the hardest part about programming is getting started.

Look at the example applications and the header examples `pacer.h`, `button.h`, `tinygl.h`, `ir serial.h`, etc.

1. Look at the on-line documentation on Learn.
2. Set up the preferences of your text editor to replace tabs with spaces and to indent by four characters.
3. Write a number of simple test applications, using one of the simple example applications (such as `updown2`) as a template.
4. Follow our coding style for ENCE260. This is documented on Learn.
5. You don't have a lot of memory (RAM or EEPROM) to play with, so keep things simple.
6. Ensure you commit your game regularly to your git repository.

## 7. Documentation

Documentation for the UCFK4 can be found at

<http://ecewiki.elec.canterbury.ac.nz/mediawiki/index.php/UCFK4>.

If you find an error, please let me know or preferably edit the documentation.