

# CM10194 Computer Systems Architecture 1 - Coursework 1

Set: Monday 21/10/2019

Due: 20:00 Monday 04/11/2019

## Learning Outcomes

By the end of this coursework you will be able to build an Arduino circuit, extend an Arduino program, and present your solution using digital media.

This coursework is done in groups of 2 students (please see below how groups are formed).

## Description

In this coursework you will be working in pairs developing a variation of the popular arcade game “Whac-A-Mole”: <https://en.wikipedia.org/wiki/Whac-A-Mole>

In short, the aim of the game is to watch some LEDs and press a button when an LED of a particular colour turns on.

The basic game uses 3 green LEDs, one white LED and a button. The green LEDs will turn on at random for a short period of time. The player’s goal is to press the button when one of them lights up. If the player presses the button in time, the white LED flashes to indicate that a point has been scored.

## Core Functionality

You have been provided with some code that turns on an LED at random and registers a button press. You will need to build a circuit to use this code, and then extend it in the following ways:

1. Check whether the button press happens at the same time as an LED. How you implement this is up to you (for example, you could use interrupts, a tight loop, or system time).
2. Add a score that is incremented if the player presses the button while an LED is on.
3. When the score reaches 10, flash all LEDs to indicate the game is over.
4. Extend to 2 player mode: Add three red LEDs and a second button to the circuit. Adapt your game so that button 1 needs to be pressed for a lit green LED, and button 2 for a lit red LED.
5. Use of a servo in a creative way (for example to show who is in the lead)

## Additional Functionality

For additional marks you should add up to three of the following extra functionalities to your game. Each additional functionality will give you extra marks (see below “Assessment” section).

- Add a third colour and button to the game.
- Vary the duration LEDs are lit and the time between them (using components that allow players to increase/decrease game difficulty).
- Include sound output (for correct/incorrect button presses).

The goal of this coursework is to have fun! Don’t be afraid to be creative and try new things - you’re building a game so think about what would make it fun to play.

## Submission

This coursework should be submitted to Moodle no later than the stated deadline. University rules apply to submissions after this deadline. Please check “Late Submission of Coursework” rules in the Comp Sci Undergraduate Programmes Handbook:

<https://moodle.bath.ac.uk/course/view.php?id=1358>

Requests for extensions should be made to the Director of Studies. Tutors cannot approve extensions.

Your submission should consist of:

1. A short (less than a minute) video with audio/written comments of your implementation, showing what it does and how. Your video should be under 30MB - if you need to compress it, you can use any software such as <http://www.squared5.com> or <http://handbrake.fr>. The .avi, .mkv or .mp4 file types are recommended.
2. All code used in your game, with appropriate commenting. These should be your .ino files, not text files (in case you use an external library, do include the code of this library when submitting your code)
3. A short report (maximum 2 pages of text) to explain your implementation and show a photo of your solution. Reports **must be in PDF format**. At a minimum, your report should include an introduction to the project, a description of how it was implemented, and how it works. For example, include how you chose to implement button presses, along with any pros and cons to this approach. Extra marks will be given for the inclusion of a schematic of your solution. You can use online tools such as <http://fritzing.org/home/>
4. A separate file named contribution (in pdf format) where you state the % of individual contributions. For example, Simba Adams: 50% Nala Clark: 50%. We would expect this to be around 50% for each member if they equally (or approximately equally) contributed for the submission. In cases where this distribution is significantly unequal (if the difference is greater than 20%), each member should write a short paragraph where they state the reasons for that. (to clarify, up to 60/40 there's no need for justification; above that (say 70-30) please each member write a short a paragraph.

You should upload these (video, code, report and contribution) as a **single zip file**.

It is your responsibility to check that your submission is correct. Once you have submitted to Moodle you should download your submission and **check it**. Submission is anonymous, so do not include your name in your code, report, video, or file names (the only exception being the content of the contribution file which we'll only read after marking the work).

## Forming your group

Both students should belong to the same allocated lab. In exceptional circumstances we will allow students from different labs to work together.

You can either choose your partner or, if you prefer, you can be allocated to work with another student.

If you would like to:

a) Choose your partner:

You and your partner should go to Moodle and sign up to the same group via the “Coursework 1 pairs sign-up” link under Coursework 1 section.

b) Be allocated to a pair:

Please complete this form <https://forms.gle/GegBMw6ySZWdVoxH7> so we know you want us to find you a partner.

If you have any problems, please email the tutors:

[teaching-cm10194@lists.bath.ac.uk](mailto:teaching-cm10194@lists.bath.ac.uk)

### Assessment

This coursework is worth 15% of your CM10194 overall mark. Your work will be assessed on three criteria (out of 100):

<b>1) Solution + Presentation</b>	<b>50 marks</b>	Solution works according to requirements; calibrated according to challenge of the problem. Demo is well executed. Shows clear understanding of the solution.
1.1) <i>Core implementation</i>	29 marks	
1.2) <i>3 Additional Functionalities</i>	21 marks	
<b>2) Code</b>	<b>25 marks</b>	Clear code. Appropriate comments, names of functions and variables. Easy to understand.
2.1) <i>Core implementation</i>	10 marks	
2.2) <i>3 Additional Functionalities</i>	15 marks	
<b>3) Mini-report</b>	<b>25 marks</b>	Clarity and conciseness of communication; appropriate presentation (written, graphic, etc). Photo of the solution. Includes reference sources if necessary.