



## MODULAR PROGRAMME

### ASSESSMENT SPECIFICATION

#### *Module Details*

|  |                                     |   |
|--|-------------------------------------|---|
| <b>Module Code</b><br>UFCETS-20-1                    | <b>Run</b><br>11SEP/1 AY            | <b>Module Title</b><br>Programming in C                   |
| <b>Module Leader</b><br>Ian Johnson                  | <b>Module Tutors</b><br>Ian Johnson |   |
| <b>Component and Element Number</b><br>B1            |                                     | <b>Weighting: (% of the Module's assessment)</b><br>13%   |
| <b>Element Description</b><br>Practical Coursework 1 |                                     | <b><u>Total Assignment time</u></b><br>5 hours + Lab Time |

#### *Dates*

|   |   |
|---|---|
| <b>Date Issued to Students</b><br>w/b 24/10/2011  | <b>Date to be Returned to Students</b><br>20th January 2012 |
| <b>Submission Place</b><br><b>PROJECT ROOM - 2Q30</b><br>(Help Desk open 9.00 - 6.00pm) | <b>Submission Date</b><br>8th December 2011                 |
|   | <b>Submission Time</b><br><b>2.00 pm</b>                    |

#### *Deliverables*

As per the attached specification.

#### *Module Leader Signature*

*Ian Johnson*

# UFCETS-20-1 – C Programming

## First Coursework - October 2011

### **Duckshoot**

Your assignment, due at 2pm on Thursday 8th December is to implement the duckshoot program as specified in the Introduction to Digital IO worksheet available on the module web page

(<http://www.cems.uwe.ac.uk/~irjohnso/coursenotes/ufs001/ufs001c1-w06-NEW.pdf>).

As per the description on the worksheet, you should start the program with alternate LEDs lit (10101010) on the MARCO RACK, and rotate this bit pattern, inverting LEDs with a trigger up, then down, then up sequence on the bottom switch.

Not "machine gunning" implies that starting with the switch up, down to fire, return to up again sequence has to be repeated in full to fire again.

Penalty for leaving the switch down means that unless the switch is returned to up in a given time, the duck reappears (or fails to disappear) to penalize the player.

In all circumstances, a single action by the user (e.g. leaving a switch up) will only cause a single change (e.g. 1 duck reappears)

The marks shown are the maximum available in each category.

### **Deliverables:**

A diagrammatic design of your program using any method with which you are familiar (10)

Printed source code:

Well commented (10)  
(-5% for C++ comments)

Program Banner (5)

Program uses functions (5)

Function banners (5)

Functions use arguments (15)

No Global variables (10)

Program is laid out in either classic C or Pascal style (10)

**Together with this page, signed off by your lab tutor**, to indicate you have demonstrated your work:

| Test   | Marks | Signature & Date |
|--|-------|------------------|
| Continuously rotating bit pattern on a suitable timebase, <b>not using a software delay loop</b> | (5)   |                  |
| Player penalty for leaving switch down (duck reappears)  | (5)   |                  |
| Not machine-gunning  | (10)  |                  |
| Direction variable by switch within game   | (5)   |                  |
| Using 2 switches to implement 4 speeds or levels of difficulty within game                       | (5)   |                  |

All the above should be submitted together with a cover sheet to the Project Room.