UHI

BEng Engineering Scheme

Module Title:	Microcontrollers and Digital Electronics UH609811
Coursework Title:	Assignment 2 : Detailed Design using a Microcontroller
Module Leader:	Lois Gray, lois.gray@uhi.ac.uk, UHI North Highland

Submission dates, word counts, and weighting:

	Latest Submission Date	Word Count	Weighting (% of module marks)
Assignment 2	25th April 2025	3000	70%

Submission as electronic files to Turnitin submission pages on VLE. 3000 words, excluding tables, diagrams, calculations, software code, pseudo-code or flowcharts, screenshot text, contents list, references list and appendices (note that appendices are not marked).

Assignment 2 assesses **LEARNING OUTCOMES 2 to 5**: The students' ability to design, implement and verify a logical solution to a microcontroller design requirement, whilst critically evaluating microcontroller firmware, software and development systems. This includes the ability to research, analyse and evaluate suitable interfacing circuits for microcontroller applications, and to function effectively as a member or leader of a team.

GENERAL INFORMATION

The submission date is the absolute latest submission date, you can submit at an earlier date. Late submissions will be penalised in accordance with UHI Academic Regulations, section 17b. Plagiarism will also not be tolerated and will be dealt with in accordance with these regulations.

1. Code re-use is encouraged, but sources of code, and of any copied or paraphrased material (paper or web-based), **must be correctly cited and referenced** in accordance with UHI requirements found at http://www.uhi.ac.uk/en/libraries/how-to. **Reports with missing citations and reference lists will fail.**
2. The assignment must be completed as teamwork and your team will submit a single report **with each individual's contribution clearly identified**. Every team member is required to write, simulate, and test some code.
3. Evidence of secure storage and version tracking is required.
4. Regardless of weighting, a pass, in accordance with UHI academic regulations, will be a minimum of 30% in all assignments, and an average mark of at least 40%.

RESOURCES REQUIRED

At the UHI learning centres delivering this module, the hardware and software required to complete Assignment 2 may be available. However, you may need to purchase the hardware/software required.

1) Arduino microcontrollers and ancillary hardware (transducers, actuators, electronic components etc) are available in the form of various development kits from online suppliers – you should discuss their availability with your lecturer. The recommended Arduino kits are:

   a) ELEGOO UNO R3 Project. The Most Complete Ultimate Starter Kit Compatible with Arduino IDE w/TUTORIAL, UNO R3 Controller Board, LCD 1602, Servo, Stepper Motor, from Amazon

   OR

   b) Arduino Starter Kit for beginner K000007 [English projects book], from Amazon

Arduino emulator/simulator software is also available from online suppliers and its procurement and installation should be discussed with your lecturer – some software can be downloaded for free e.g. TinkerCAD Circuits from https://www.tinkercad.com/learn/circuits or Virtronics at http://www.virtronics.com.au/Simulator-for-Arduino.html (for the latter, functionality becomes restricted after 7 days unless you pay for a licence).

Alternatively, you may use a different controller or microcontroller, such as a Programmable Logic Controller (PLC), Field Programmable Gate Array (FPGA), or Industrial Internet of Things (IIoT) equipment. Recommendations for these are:

2) Siemens Logo 8 with Logo Soft Comfort IDE

3) Xilinx Spartan 7 FPGA (e.g. Arty S7) with Xilinx Vivado Design Suite

4) National Instruments LabVIEW Virtual Instrumentation System

ASSIGNMENT 2: BRIEF

TEAM AIM: To design a closed loop Proportional-Integral-Derivative (PID) control system for an industrial application, implementing the control aspect with a microcontroller, PLC, FPGA, or IIoT equipment.

In this assignment you are required to work in teams of two or more. Marks are allocated for teamwork, per the Marking Scheme on page 6.

System monitoring and control is a very common application for computer systems. In this coursework your team will design a control system which monitors one or more inputs and uses these in a closed-loop PID control algorithm to perform a typical control application.

You may choose any suitable application. Some ideas are:

- Control of a variable speed drive, e.g. a conveyor belt or fan,
- Control of a process variable, e.g. temperature, pressure, or flow,
- Automatic Test Equipment (ATE).

Your team will design, simulate, construct, test and evaluate their system, then demonstrate working hardware to your tutor. The entire process will be documented in **a single** team report, **with individual contributions clearly identified**.

The design objectives are:

a) Draw a high-level block diagram of your team's chosen system. This must include sensors, actuators, feedback, a digital PID controller, and a display (to monitor functionality and facilitate fault finding).

b) Identify what hardware, firmware, programming language, development and storage environments are required for each block. Justify your team's choices in terms of:

- **Technical capability**: voltage, current and power requirements, range, accuracy, sensitivity, resolution, operating/conversion speed, interfaces, etc.

- **Useability**: tolerance, environmental operating conditions, reliability, complexity, optimisation and programming features, simulation capability, etc.

- **Availability**: cost, lead time, manuals and help availability, physical size, etc.

- **Safety and Security**: security features, electromagnetic capability, risks and hazards (to people and equipment), mitigation of risks and hazards, software storage and version control, data protection, downloading/sharing/uploading integrity, bug and breach logging methods and capability, etc.

c) Use the knowledge obtained in (b) to write a detailed design specification for your team's system.

d) Source the required hardware, firmware, and development environment. You may purchase additional hardware if you require it, e.g. from https://coolcomponents.co.uk/

e) Write interface specifications for each of the interfaces as defined by your team's block diagram (e.g. what is the sensor's output range, what pin is it driving, what is the input range of this pin, what is the actuator's drive requirements, what is required to source this drive, etc.). **Every team member must individually contribute to this.**

f) Design and test appropriate analogue signal processing circuitry (as required) for each interface defined in (e). **Every team member must individually contribute to this.**

g) Using a recognised software development tool (e.g. flow chart, Unified Modelling Language (UML), or pseudo-code) design the necessary code to implement the full system. **Every team member must individually contribute to this**. The code should:

   i. Drive the actuator via appropriate signal conditioning.
   ii. Monitor the actuator feedback signal via appropriate analogue signal conditioning hardware and digital signal conditioning code.
   iii. Monitor the reference demand input via appropriate analogue signal conditioning hardware and digital signal conditioning code.
   iv. Calculate an error value from (iii) – (ii).
   v. Implement a PID control algorithm, including anti-windup.
   vi. Display and log any outputs' which provide functionality information, to facilitate fault finding.

   You may base your code on any suitable examples. These are usually available via Integrated Development Environments (IDEs) and software providers' websites (e.g. Arduino IDE and website, TinkerCAD and EDA Playground websites, etc.).

h) Write a program, to implement all the actions defined in (g). **Every team member must individually contribute to this**.

i) Fully simulate the design using, if possible, both functional and timing simulation. Document the simulator operation using screenshots and show evidence of advanced testing techniques, such as use of break-points, and debug message printing. **Every team member must individually contribute to this.**

j) Upload your team's program/programs to hardware and test the complete system (hardware and software) for various input parameter settings – **your team should design a formalised test procedure to accomplish this**.

k) If necessary, modify the code and repeat (h) to (j) as required.

l) Critically evaluate your team's final design results in terms of compliance to the specification defined in (c). **Every team member must individually contribute to this.**

m) As a team, demonstrate the full working system to your tutor.

n) Write a critical reflection of how your team worked together. **Every team member must individually contribute to this**.

BEng Engineering Scheme	
Module Title:	Microcontrollers and Digital Electronics UH609811
Coursework Title:	Assignment 2 : Detailed Design using a Microcontroller
Module Leader:	Lois Gray, lois.gray@uhi.ac.uk, UHI North Highland

MARKING SCHEME

This assignment is worth 70% of the overall module marks. Additional marks will be given for solutions that include advanced features, such as auto-calibration, auto-ranging, averaging etc.

Mark are awarded as a group for common parts (Introduction, Justification, System Specification, Test, and Conclusion) and individually for all other parts (Interfaces, Analogue processing, Digital design, Simulation, Evaluation, and Team Working).

Note that if your contribution to an individually marked part is not obvious, you will get 0 marks for that part.

Marking Scheme	Marks
(a) Introduction and Block Diagram	5
(b) Hardware, firmware, programming language, development and storage environments: research and justification	5
(c) System Specification	5
(e) Interface Specification	5
(f) Analogue signal processing design, test and evaluation	5
(g) and (h) Digital design and coding	
Secure storage, version tracking, and optimisation	5
Sensors (inputs) signal processing	5
Actuators (outputs) signal processing	5
Controller	5
Display driver and logging	5
Advanced features	5
(i) Software simulation	10
(j) Practical system test	10
(l) System evaluation	10
(m) Conclusion and demonstration	5
(n) Teamworking: Role allocation, tracking of team progress, ensuring equal contribution from team members	10
Total	**100**

ASSIGNMENT 2: SUGGESTED REPORT STRUCTURE

The following notes are **suggested** topics you should consider answering or covering in your report. It is not expected that you cover all of the points listed, but you should certainly cover some of them. Word counts are very approximate.

(A) INTRODUCTION AND BLOCK DIAGRAM (150 WORDS)

- What is the aim of this coursework and what does your team hope to achieve?
- What does this report cover?
- What type of control system has your team chosen. Why?
- What are the advantages and disadvantages of using digital control for this application?
- How might development environments assist in rapid application development?
- What benefits are offered by different programming languages?
- What are the roles of each team member?

(B) CAPABILITY AND JUSTIFICATION (250 WORDS)

- What are the key capabilities of chosen hardware, firmware, programming language, development and storage environments? Are these good, acceptable or poor compared to your team's expectations? Why?
- How might the functionality of the final system be affected by the parameters of chosen hardware and firmware (e.g. sensitivity and accuracy of the sensors and actuators, resolution of ADC and DAC, type of controller)?
- How might the functionality of the final system be affected by the parameters of chosen software language, and development and programming environments (complexity, speed of operation, reliability, code checking, optimisation, simulation and debugging capability, programming functionality and security)?
- What are the risks and hazards associated with this mini-project? How will your team mitigate these?
- What security, version tracking, and logging methods is your team using? Why?
- Descriptions of operation (e.g. of sensors, actuators, controller etc.) and relevant datasheet excerpts should be included as appendices. All appendices should be referred to in the main text.
- Use citations and tables in this section.

(C) SYSTEM SPECIFICATION (200 WORDS)

- What parameters are important for correct system operation? Why?
- In terms of the control system, what steady state error is tolerable? What transient response is tolerable? Over what ranges of inputs and outputs?
- What tolerances has your team allowed for? Why? Over what input/output range?
- What environmental conditions will the system work in?
- Are there any other constraints, e.g. size, life, ease of maintenance (for fault finding)?
- What optimisation is required (space, power, size, speed, security)?
- What security, version control, and logging measures are relevant? Why?
- Use citations and tables in this section.

(E) INTERFACE SPECIFICATION (200 WORDS)

- What are the required supplies for each device (AC, DC, voltage, current, tolerance)? Can the controller supply these or are external power sources required?
- What type of signals does each sensor produce? What are the key specifications of these signals (e.g. AC, DC, digital, analogue, voltage, current, frequency, waveform, resistance, impedance)?
- Which pins or components are your team using for sensor inputs? Why?
- What type of signals does each actuator and display require? What are the key specifications of these signals (e.g. AC, DC, digital, analogue, voltage, current, frequency, waveform, resistance, impedance)?
- Which pins or components are your team using for actuator and display driving? Why?
- Use tables and citations in this section.

(F) ANALOGUE SIGNAL PROCESSING (200 WORDS)

- What sensor analogue signal conditioning circuitry is required to allow connection to the controller?
- What actuator and display drive circuitry is required to allow connection to the controller?
- What signal limits apply to prevent damage? Has signal limiting being applied? How?
- Does the circuitry limit or degrade the operation in any way? How?
- How has your team tested the circuitry? How?
- Does is do what it is designed to do? Why/why not?
- Use circuit diagrams and results tables in this section.

(G)AND (H) DIGITAL DESIGN: SECURITY AND SAFETY (200 WORDS)

- What security measures has your team implemented (storage security, code security, programming security, etc.?) How?
- Are these security measures adequate? Why/why not?
- What version control measures has your team implemented? How?
- Are these version control measures adequate? Why/why not?
- What safety measures have your team implemented?
- Are these safety measures adequate? Why/why not?
- What optimisation has your team implemented? Why?

(G)AND (H) DIGITAL DESIGN: SENSORS (200 WORDS)

- Flow chart, pseudo-code, or UML for this section.
- What Analogue to Digital Conversion (ADC) technique has your team used to read the sensor signals? How is this implemented?
- What functions/procedures are implemented in the software to handle the ADC output?
- What errors might be present in terms of resolution and sample-rate of the ADC used?
- What Digital Signal Processing (DSP) techniques has your team implemented and why?
- How do the DSP techniques modify and improve the raw data from the sensor?
- What software functions/procedures has your team used to achieve this?
- Has your team implemented any additional DSP to further enhance the quality/usability of the raw data (e.g. filtering)?

(G)AND (H) DIGITAL DESIGN: ACTUATORS (200 WORDS)

- Flow chart, pseudo code, or UML for this section.
- What Digital to Analogue Conversion (DAC) method is used to drive the actuator?
- By what method is feedback from the actuator obtained?
- How are the DAC and feedback implemented?
- What functions/procedures are implemented in the code, to process the signal to the DAC?
- What functions/procedures are implemented in your software to process the feedback from the actuator?
- What errors might be present in terms of resolution?
- Has your team implemented any additional DSP to further enhance the quality/usability of the actuator or feedback from it (e.g. averaging)?

(G) AND (H) DIGITAL DESIGN: CONTROLLER (200 WORDS)

- Flow chart, pseudo code, or UML for this section.
- What type of controller has been designed and how has this been implemented?
- What are the optimum steady state and transient responses? Why? Use graphs.
- Is the controller calibrated or tuned for optimum response? How?
- What additional control features have been implemented and for what benefit?

(G) AND (H) DIGITAL DESIGN: DISPLAY AND FUNCTIONALITY LOGGING (200 WORDS)

- Flow chart, pseudo code, or UML for this section.
- What type of display/output has your team used and why?
- What functions/procedures are used in the software to display the processed data via the display chosen? How do these work?
- What signals (outputs and/or register values) are being logged? Why?
- How does the logging system work?
- Has your team used any additional features, beyond simply displaying the output, that enhance the functionality of the display in any way?

(I) SIMULATION (250 WORDS)

- How did your team test the functionality of your software via the simulator?
- What were the desired results of each of the tests that were carried out?
- How did your team diagnose faults/problems at each stage (compilation, assembly, simulation, emulation)?
- With reference to appropriate screenshots, show each test process and result achieved.
- Was an emulator used? What are the advantages of using an emulator?
- What changes did your team have to make to the code to fix problems/enhance functionality/improve performance as a result of simulation testing. What results led to these changes and were the changes successful?

(J) SYSTEM TEST (250 WORDS)

- What hardware tests were carried out?
- How did these tests prove that the system met the specification?
- What were the results of these tests?
- What observations are evident from these tests?

- How did the results of hardware tests align with the results of code simulation? What caused any differences and how did this affect testing and subsequent code changes?
- What changes did your team make to the code to fix problems/enhance functionality/improve performance as a result of hardware testing. What results led to these changes?
- Use graphs, tables and diagrams in this section.

(L) SYSTEM EVALUATION (250 WORDS)

- How well does the final system meet the specification?
- What errors have been noted?
- What is the cause of these errors?
- What further development of the system hardware could be undertaken to address any errors identified?
- What further development of the system software could be undertaken to address any errors identified?

(M) CONCLUSION (100 WORDS)

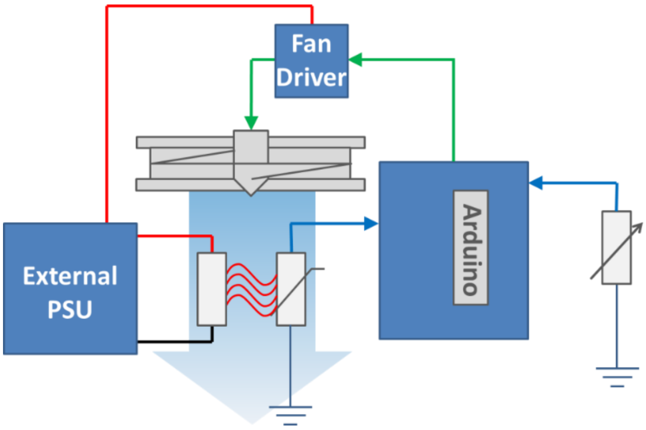
- Has the intention of this experiment/design exercise been met in your team's opinion?
- What new knowledge have individual team members gained from conducting this experiment/design exercise?
- How relevant does your team think this experiment/design exercise is to the implementation of practical, "real-world" environmental-measurement/data-acquisition systems?

(N) TEAM WORKING REFLECTION (150 WORDS)

- How did you allocate team roles? Did you have to change these? Why/why not?
- What were your main communication methods? How effective was communication between team members?
- How effective was the team in tracking and actioning progress?
- What did you do well as a team? What did you do badly as a team?
- Did every member contribute equally?
- Did any member let the team down significantly? How did your team cope with this?
- Did any member shine, e.g. as a team leader, communications co-ordinator, workshop supervisor, or action chaser?
- What would you change if you were to do this assignment again?

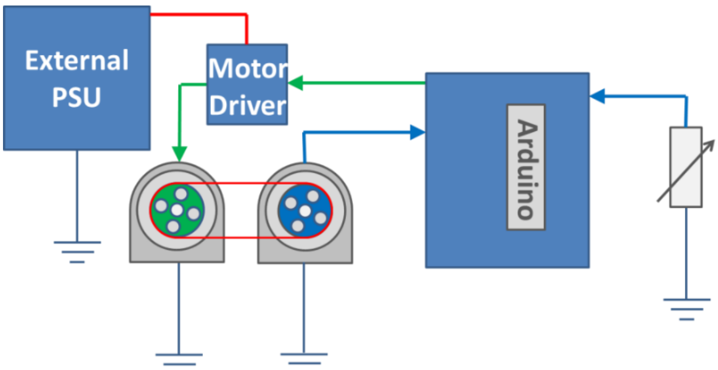
SOME SUGGESTED SYSTEM CONFIGURATIONS

TEMPERATURE CONTROLLER EXPERIMENT:



An external PSU is used to heat a low-ohm resistor placed in close proximity to a thermistor (e.g. LM35). A standard 80mm PC cooling fan is placed such that it blows air over the resistor/thermistor arrangement and is driven, via suitable driver circuitry (i.e. a medium power BJT with back-emf protection), from the Arduino. A **temperature** set-point is applied via an external potentiometer – remember to limit the current supplied by this potentiometer to avoid damaging the Arduino.

DC MOTOR CONTROLLER:



Two small d.c. motors are fixed in position and linked via pulley-wheels and an elastic-band. The voltage across the terminals of one is taken as an input to the Arduino whilst the other is driven (typically using PWM) from the Arduino, via suitable driver circuitry (as above). An external PSU is used to avoid over-driving the Arduino 5V line. A **speed** set-point is applied via an external potentiometer – the same current precautions as above should be observed.