PHC 435: Pharmaceutical Data Acquisition and Analysis
PHM 1138: Electronics for Pharmaceutical Applications

I. CONTACTS

INSTRUCTORS:

Dr. David Dubins
Course Coordinator
d.dubins@utoronto.ca

STUDENT HOURS:
Mondays, 10a–12p EDT in PB860 (Optional Tutorial: 12p–1p)
Wednesdays, 10a–12p EDT in PB450 (Optional Tutorial: 12p–1p)

II. COURSE OVERVIEW

Welcome to PHC435/PHM1138! The goal of this course is to introduce students to theoretical and applied concepts in electronic circuitry, for the purpose of collecting and analyzing experimental data in pharmaceutics and other contexts. The course is designed as approximately two thirds small-group on-line didactic teaching, and one third on-line virtual/practical exercises to experiment with and illustrate concepts. The course discusses introductory circuit design, with an emphasis on how common components work (e.g. resistors, capacitors, diodes, transistors, operational amplifiers, and a variety of sensors) in scientific and pharmaceutical manufacturing instrumentation.

Practical and mathematical aspects of circuit design are discussed (e.g. Ohm's Law, voltage dividers, analog vs. digital signals). There is a heavy emphasis on programming in C++, taught from an introductory level, which will complement learning activities. Assessments will include quizzes, problem sets, a design project, a participation component, and a final test.

With the recent advent of low-cost, consumer-level microprocessors (e.g. ATtiny, Arduino, Raspberry Pi, ESP8266, ESP32), affordable and accessible processing power has empowered researchers with resources to take experimental designs to new heights. Such microprocessors are relatively simple compared to the complexity of today's desktop computer; however, are more than powerful enough and fast enough to control sophisticated equipment such as scientific instrumentation and 3D printing. Previously, DACs (Digital-to-Analog Converters) were thousands of dollars, requiring high programming aptitude to bridge the gap between computer and instrument. Serial communication ports were reliable only at slower speeds (e.g. 1200 bps). Serial communication was finicky, and required access to equipment subroutines not always readily available. However, the climate has now changed for experimental design. Libraries are readily available, interfaces are more intuitive, and a large open-source community exists to support scientists and hobbyists alike. Knowledge of programming and circuitry will provide a solid foundation not only in experimental design and analysis for this field, but in many other areas as well.

The modern era of electronics has caused a paradigm shift. Due to economies of scale, electronic components have become very inexpensive. The electronics hobbyist niche has driven the development of modular electronic components marketed for general purposes, geared towards on open-source platforms (e.g. opto-isolator power relay circuits, H-bridge motor controllers, and frequency-matched RF transmitters/receivers). Circuits that would previously need to be thoughtfully considered and designed are readily available and packaged as low-cost, ready-to-use modules. This course will examine some of these modules and their usefulness in circuit design.

STUDENT LEARNING OUTCOMES:

By the end of PHC435/PHM1138, students will …

1) Be able to properly interpret other people's circuit diagrams and draw their own diagrams in a manner that will enable them to accurately record and share their work with others. Students will be able to draw circuit diagrams with enough detail so that others can build the same project. They will develop a foundation for "circuit diagram literacy".
2) Develop a working knowledge of some basic building blocks in electronics: resistors, capacitors, relays, transistors, MOSFETs, motors, op-amps, voltage regulators, etc. Understanding how electronic components fit together will enable you to go beyond following other people's circuit diagrams, synthesizing their own modules and ideas.
3) Become proficient in C++ programming, which is a wonderful segue into learning other programming languages. Once students learn how to code in one language, learning another is often a matter of translation, which is much faster to pick up (e.g. Python, Matlab®, and R-project). Students will learn the basics of writing a computer program. The course has many opportunities to write and compile new code. Students will learn proper programming etiquette, including commenting, selecting and working with appropriate variable types, writing subroutines, functions, and declaring local and global variables. After students are finished this course, they will be able to write, compile, and upload code for the Arduino Uno's microprocessor, the ATmega328, using the open-source Arduino IDE. These skills are transferrable to programming other microprocessors.
4) Be able to design, create, assemble, and test their own projects and scientific equipment, from raw sensor output to data filtering and logging.
5) Be able to develop and refine their troubleshooting skills, and become more confident in their circuit building and testing abilities. Students will be empowered to diagnose and solve their own problems. They will recognize the value and complexity in commercial electronics around them.
6) By learning the fundamentals, students will no longer perceive an electronic device as a single black box that either works or doesn't work, but rather as a sum of parts that can be repaired, and failing that, at least salvaged for working parts.
7) Be able to design and build circuits safely and carefully, always being mindful of the dangers of high voltage.

8) Have fun with electronics and programming!

TEXTBOOK AND RESOURCES:

Required Text

Dubins D. Electronics and Microprocessing for Research 2nd Edition: You Can Make It. Cambridge Scholars Publishing, Newcastle upon Tyne, UK. November 1, 2019.

The course textbook was written specifically for this course.

Recommended Texts

Scherz P., Monk S. Practical Electronics for Inventors, Fourth Edition. McGraw Hill, New York New York, USA, March 24, 2016.

Gibilisco, S. Beginner's Guide to Reading Schematics, Fourth Edition. McGraw-Hill Education. August 8, 2018.

Course notes, objectives, and information will be made available through the University of Toronto Quercus system, and the course website: http://pb860.pbworks.com. A printable handout will be provided for each course section, which will also include a description of the practical activities.

III. HOW THE COURSE IS ORGANIZED

COURSE SCHEDULE AND LOCATION

Section	Description	Mondays (P860) & Wednesdays (PB450) Lectures: 10a–12p Tutorials: 12–1p
1	Introduction to Electricity.	M 11-Sep-23
		W 13-Sep-23
		M 18-Sep-23
2†	Power, Logic, and Capacitance.	W 20-Sep-23
		M 25-Sep-23
3	Introduction to programming in the Arduino C++ Environment.	W 27-Sep-23
		M 02-Oct-23 (in PB450)
		W 04-Oct-23
4†	Arduino Pins, and Functions.	W 11-Oct-23
		M 16-Oct-23 (in PB450)
		W 18-Oct-23
5	Switching Higher Power Devices: Relays, Transistors, MOSFETs, TRIACs.	M 23-Oct-23

		W 25-Oct-23
6†	Process Control.	M 30-Oct-23 ***Design Project Proposals Due***
		W 01-Nov-23
7	Operational Amplifiers.	M 13-Nov-23
		W 15-Nov-23
		M 20-Nov-23
8†	Data Filtering, Data Smoothing, Data Logging.	W 22-Nov-23
		M 27-Nov-23
		W 29-Nov-23
DESIGN PROJECT BLOCK		M 04-Dec-23
		W 06-Dec-23
		T 07-Dec-23 ***Design Project Presentations: 11:00a-12:00p*** ***Final Design Project Due by 5p***
	Final Examination	To Be Announced (Will take place during Final Assessment Period)
		Shaded dates: Practical Activities planned during these lectures.

†Problem set handed out at the end of this section (due at the beginning of class the following week, unless otherwise indicated).
For important academic sessional dates, visit:
https://www.artsci.utoronto.ca/current/dates-deadlines/academic-dates
https://sgs.calendar.utoronto.ca/sessional-dates

LECTURES

Lectures will be taught using a combined approach of didactic teaching, simulated practical exercises, and physical circuit building. Typically, the first part or parts of each section will be hosted in the form of small group didactic teaching. The third hour will be used for practical exercises, aimed at applying the key concepts and seeing them in action. Design of this course therefore resides on all levels of Bloom's taxonomy: remembering, understanding, applying, and creating. The practical component of the course is fundamental for students to connect with lecture objectives, and consequently, participation is an important part of assessment.

The Arduino platform (specifically, the Arduino Uno R3, programmed by the open-source Arduino IDE) will be used in many lectures to control and read from scientific equipment. Some activities will require the use of a virtual circuit building platform (Tinkercad.com), and some will require physically building and testing circuits, by breadboarding them.

In terms of programming, the course has a heavy teaching component of programming in C++, an important and ubiquitous computer language in the sciences. The course does not assume any prior experience with computer programming.

TUTORIALS (OPTIONAL)/OFFICE HOURS

Many practical activities have been planned for this course. The completion of these activities will contribute to your participation mark of the course. It is expected that these activities will be completed during the allotted class time.

Sometimes, troubleshooting a project can take much more time than anticipated. In order to provide extra practical time for planned activities, **optional tutorials** have been scheduled after each lecture. This will allow a student who requires more time or assistance with the planned activities the benefit of some additional guided feedback and project troubleshooting. The optional tutorials will also provide extra opportunities to gain activity participation marks.

These optional tutorials can be used to discuss any aspect of the course, to work on incomplete activities from previous lectures, and can also be used as office hours.

HOW TO PREPARE FOR CLASS

1) Read the associated section in the Course Textbook.
2) Print out the section handouts.
3) For classes that include practical activities (shaded dates in the Course Schedule above), locate and prepare the necessary components. Each section slide pack will begin with a slide indicating which components to get ready for that section.
4) In general, most lectures and practical activities will require:
   - Laptop/Notebook
   - Arduino Uno + USB cable (provided on loan by course instructor)
   - Course Textbook
   - Ruler
   - Pencil
   - Highlighter

IV. GRADING SCHEME

The final grade will be based on four problem sets, two on-line quizzes, one final test, and class participation. Assessment methods are balanced to address different learning styles. Questions on the problem sets will be drawn primarily from the lecture material, course textbook, and concepts discussed in lectures and activities.

The schedule and weighting of the evaluative components of the course are as follows:

1) Problem Sets (4)

Weight: **40%**
Best of 4 problem sets worth 16%, the other three worth 8%.
Dates: Provided in the course schedule.
Due: One week from assignment of the problem set.

2) Quizzes (2)
Weight: **10%**
Dates: Surprise in-class quizzes.

3) Design Project *(at home, and during tutorials)*
Weight: **15%**
Dates: Design Project Proposal: 30-Oct-23
Final Design Project: 7-Dec-23

4) Final Exam (3 hours)
Weight: **25%**
Dates: To be scheduled during the final examination period.
Material: Entire Course.
Assessment style: similar to problem sets. This will be an end-of-term, timed final assessment. Coding is on the final test.

5) Class Participation
Weight: **10%**
Based on: participating in class discussions, completing simulated and practical activities, and showing initiative during class. Participation of each exercise will be confirmed in person or by screen sharing for any online components (Tinkercad/Arduino IDE), and by webcam where appropriate (circuit building). Completion of each exercise will be logged by the instructor, and will constitute a portion of the participation grade.

The graduate version of this course (PHM1138) will have additional questions in assessments (problem sets and final test).

Design Project
A design project will involve students designing programming, and testing their own circuits. Design projects are to be performed ***individually*** by students, and will integrate concepts learned throughout the course. The goal of the design project is to propose, synthesize, and optimize, and program a new circuit. The function of the circuit should relate to course objectives. Proposals for individual design projects are due at the beginning of Section 6 (30-Oct-23), and will constitute a portion of project evaluation. Design projects are due at the end of the course. A final report is handed in with the project, including a description of the purpose and functionality of the circuit, a detailed circuit diagram, a copy of the final sketch, and a testing protocol with experimental results. Any calculations required to design the circuit should be provided as an appendix. Further details regarding evaluation of the design project are provided in the laboratory manual. Students may work on their design projects at home, and may also bring their projects to work on during tutorials.

V. COURSE POLICIES

Lateness Policy
Problem sets are due one week after being assigned, by the **beginning** of class, by submitting through Quercus. Problem sets handed in at the end of class will be considered late. For late submissions of course assessments, there will be an academic penalty imposed of 3% per calendar day. Submissions will not be accepted beyond 1 week from the original due date.

E-mail

• For course concerns or issues with non-academic problems, such as conflicts, illness and academic accommodations, please email d.dubins@utoronto.ca.

• When you e-mail an individual, the language and tone of your email professional. Email only one member of the teaching team. Most emails will receive a reply within 24 hours of being sent (except on weekends) but keep your expectations reasonable as to the degree of detail that an email reply to your enquiry can realistically provide.

Course Environment

• The University of Toronto is committed to equity, human rights and respect for diversity. All members of the learning environment in this course should strive to create an atmosphere of mutual respect where all members of our community can express themselves, engage with each other, and respect one another's differences. U of T does not condone discrimination or harassment against any persons or communities.

Privacy Policy

• Parts of this course, including your participation, will be recorded on video and will be available to students in the course for viewing remotely and after each session.

• Course videos and materials belong to your instructor, the University, and/or other sources depending on the specific facts of each situation and are protected by copyright. Do not download, copy, or share any course or student materials or videos without the explicit permission of the instructor.

• At times, you will be required to share your desktop, and turn on your webcam to share your completed (working) activity. This will be required in order to assess your participation mark, and help you troubleshoot your practical activities.

Absences

Attendance of each lecture is mandatory and will be recorded.

Guidelines for Declaring Absences in Acorn:

https://www.artsci.utoronto.ca/current/academics/student-absences

- **ACORN Absence Declaration Tool** to declare an absence once per academic term (e.g., the fall term) for a maximum period of seven (7) consecutive calendar days. The seven-day declaration period can be retroactive for up to six (6) days in the past, or proactive, up to six (6) days in the future.
- A Verification of Illness Form, College Registrar's Letter, or Letter of Academic Accommodation from Accessibility Services

You must also advise Dr. Dubins, the course coordinator, of your illness to receive consideration. Students must attend and complete at least 80% of the in-person periods (typically 20 out of 24) to complete PHC435.

VI. TECHNOLOGY REQUIREMENTS

REQUIRED EQUIPMENT

- **A laptop computer** is required for this course.

OPTIONAL EQUIPMENT

- Students may purchase an **Arduino Uno Starter Kit**, which will include the Arduino Uno r3 MCU, a stepper motor, a servo motor, a breadboard, jumper wires, LEDs, resistors, a thermistor, and various other components. The cost of this kits range from $30-60 CDN plus taxes. Here is one such kit:
https://www.amazon.ca/Elegoo-Project-Starter-Tutorial-Arduino/dp/B01D8KOZF4
- You may also purchase additional equipment/components for your design project.

Further advice may be given at the first class depending on availability of these kits.

This course requires the use of computers, and technical issues are possible. When working on an assignment, students are responsible for scheduling enough time to allow for reasonable delays due to technical difficulties to be overcome, so such issues will not be acceptable grounds for deadline extension. Particularly, maintaining an up-to-date independent backup copy of your work is strongly recommended to guard against *e.g.* hard-drive failures, corrupted files, lost computers, etc.

VII. INSTITUTIONAL POLICIES AND SUPPORT

ACADEMIC INTEGRITY

Academic integrity is essential to the pursuit of learning and scholarship in a university, and to ensuring that a degree from the University of Toronto is a strong signal of each student's individual academic achievement. As a result, the University treats cases of cheating and plagiarism very seriously. The University of Toronto's Code of Behaviour on Academic Matters (https://governingcouncil.utoronto.ca/secretariat/policies/code-behaviour-academic-matters-july-1-2019) outlines the behaviours that constitute academic dishonesty and the processes for addressing academic offences. Potential offences include, but are not limited to:

In practical work:
1. Using someone else's ideas or words without appropriate acknowledgement.
2. Submitting your own work in more than one course without the permission of the instructor.
3. Making up sources or facts.
4. Obtaining or providing unauthorized assistance on any assignment.

On tests:
1. Using or possessing unauthorized aids.
2. Looking at someone else's answers or collaborating or discussing answers during an exam or a test.
3. Misrepresenting your identity.

In academic work:
1. Falsifying institutional documents or grades.
2. Falsifying or altering any documentation required by the University.
3. Sharing solutions to the online homework.

In computer programs:
1. Not properly referencing included libraries.

2. Not properly referencing any code used to help generate your code.
3. Copying an algorithm or snippet of computer code without referencing the source and author
4. Copying someone else's program, modifying it to make it look different, and submitting it as your own work.

Course Policy on Use of Artificial Intelligence Tools (e.g. ChatGPT):

The goal of technical writing is to report clearly (easily understood), accountably (accurate and honest reporting), and transparently (not trying to hide or obscure data or procedural errors). Artificial intelligence tools can assist in this regard, provided the tools are used in ethical ways:

- Students ***may not*** use artificial intelligence tools for taking tests in this course.
- Students ***may not*** use artificial intelligence tools for creating computer code to hand in for evaluation. However, these tools may be useful when gathering information from across sources and assimilating it for understanding.
- Students ***may*** use generative AI tools for technical writing aspects of assignments, to improve the quality and clarity of written work (*e.g.* to assist in polishing writing in the design project proposal or final report).
- Any *specific* content produced by an artificial intelligence tool ***must be cited appropriately***. Many organizations that publish standard citation formats are now providing information on citing generative AI (e.g., MLA: https://style.mla.org/citing-generative-ai/).

    e.g.: "Using clear scientific language, describe the importance of surfactants in liquid drug formulations. Add specific real-world examples, elaborate on mechanisms, and discuss current challenges." prompt. ChatGPT-3.5, OpenAI, 7 Sep. 2023, https://chat.openai.com.
- **Students are ultimately accountable for the work they submit.**
- If you have any questions about the use of AI applications for course work, please speak with the instructor.

All suspected cases of academic dishonesty will be investigated following procedures outlined in the Code of Behaviour on Academic Matters. If you have questions or concerns about what constitutes appropriate academic behaviour or appropriate research and citation methods, you are expected to seek out additional information on academic integrity from your instructor or from other institutional resources (see https://www.academicintegrity.utoronto.ca/).

VERIFICATION OF ACADEMIC ORIGINALITY: Normally, students will be required to submit their course essays to the University's plagiarism detection tool for a review of textual similarity and detection of possible plagiarism. In doing so, students will allow their essays to be included as source documents in the tool's reference database, where they will be used solely for the purpose of detecting plagiarism. The terms that apply to the University's use of this tool are described on the Centre for Teaching Support & Innovation web site (https://uoft.me/pdt-faq).

COPYRIGHT

Students with diverse learning styles and needs are welcome in this course. The University of Toronto is committed to accessibility: if you require accommodations for a disability, or have any other accessibility concerns about the course, please contact Accessibility Services as soon as possible.

ACCOMMODATIONS FOR RELIGIOUS OBSERVANCES

Following the University's policies, reasonable accommodations will be made for students who observe religious holy days that coincide with the due date/time of an assignment, tutorial, class or laboratory session. Students must inform the instructor **before** the session/assignment date to arrange accommodations.

ACCESSIBILITY NEEDS

Students with diverse needs are welcome in this course. The University of Toronto is committed to accessibility: if you require accommodations for a disability, or have any other accessibility concerns about the course, please contact Accessibility Services as soon as possible.

ADDITIONAL SERVICES & SUPPORT

The following are some important links to help you with academic and/or technical service and support:

- General student services and resources at Student Life
- Full library service through University of Toronto Libraries
- Resources on conducting online research through University Libraries Research
- Resources on academic support from the Academic Success Centre
- Learner support at the Writing Centre
- Information for Technical Support/Quercus Support

ACKNOWLEDGEMENT OF TRADITIONAL LANDS

We wish to acknowledge this land on which the University of Toronto operates. For thousands of years, it has been the traditional land of the Huron-Wendat, the Seneca and, most recently, the Mississaugas of the Credit River. Today, this meeting place is still the home to many Indigenous people from across Turtle Island and we are grateful to have the opportunity to work on this land.