Electronics and Microprocessing for Research

Electronics and Microprocessing for Research:

You Can Make It

By

David Dubins

Cambridge
Scholars
Publishing

Electronics and Microprocessing for Research: You Can Make It

By David Dubins

This book first published 2018

Cambridge Scholars Publishing

Lady Stephenson Library, Newcastle upon Tyne, NE6 2PA, UK

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN (10): 1-5275-1333-5
ISBN (13): 978-1-5275-1333-4

To my sons Aidan and Remy, for inspiring and testing my projects;

To Andrew Cooper, for sparking my interest in electronics;

and to Lily, for supporting many late nights in the lab.

Epigraph by Remy Dubins

TABLE OF CONTENTS

List of Figures

List of Tables

ACKNOWLEDGEMENTS

Many people assisted in the inception and development of this text.

Dr. Robert B. Macgregor, Jr. inspired this work by encouraging me to turn a fixed USP dissolution apparatus into an undergraduate course, and he helped me launch it within a short time span of six months.

Andrew Cooper taught me much of the tacit and practical knowledge, for example, *Practical Strategies to Reduce Signal Noise* (Table 8-5) and providing a virtual ground for a probe (Figure 7-31).

Dr. Will Cluett kindly reviewed and provided comments for Section 6.

There are a number of students who in taking this course have helped to refine and improve this work immensely: Celeste Vicente, Brittany Epp-Ducharme, and Nabeel Tariq. Your enthusiasm and keen eyes for detail are warmly appreciated.

Lastly, I would like to thank the institution in which I work, the Leslie Dan Faculty of Pharmacy at the University of Toronto for providing a positive and supportive educational environment and infrastructure, and plenty of batteries.

PREFACE

It's not *supposed* to work.

One of the things I hope you will discover throughout this course is that we take our technology for granted. This attitude is deeply embedded in our consumer culture. As end users, we are intentionally blinded to failures during product development. We demand quality, taking good design for granted. We purchase the latest gizmos at our favourite technology stores, and after a year or two of using them, either they break and we buy new ones, or we toss them aside because we are bored and done with them, wanting newer gizmos. We are *accustomed* to the idea that a device is supposed to work. Why shouldn't it? After all, we paid good money for it. It comes in a box, with a guarantee. If it doesn't work, we get a refund after a flustered sales person squints apologetically at the original store receipt.

These gizmos *work* because of careful design, quality parts, automated assembly processes, meticulous QC checks, and market research—something that engineers, scientists, manufacturers, and business people spend their entire careers working on. As a novice then, you will very likely become frustrated early on that the circuit you are building does not light up an LED, measure the temperature, or send text to a screen. Perhaps little components will produce strong-smelling smoke. Although a lay person might call this "broken", "screwed up", or "horrible" and feel discouraged about their abilities, this frustration is more appropriately called the *design and testing* phase–where things necessarily won't work. Why should they? The circuit doesn't even exist yet, let alone function flawlessly enough to sell in a store.

As a designer then, it is your job to put on an optimistic hat, and sharpen your troubleshooting skills. I can promise you that during the activities in this course, you will hit a dead end where you feel that you have tried everything you can, yet your circuit still doesn't work. You might feel frustrated, discouraged, or defeated. I can promise you this because it happens to me all the time.

However, I can also promise that if you hang in there, when you finally do figure out a critical piece of the puzzle that makes your project spring to life, all of that frustration will dissipate (after the urge to kick yourself passes). It will be replaced by a feeling of relief, satisfaction, pride, happiness, and perhaps the thrill that addicted me to this area of study in the first place. Electronics are SO COOL!!!

SECTION 0

INTRODUCTION AND COURSE OBJECTIVES

Introduction

We are taught at a very early age to start counting at 1. However, with computer programming in microprocessing, we need to get into the habit of starting to count from zero. It is only fitting then to begin the introduction of this course with Section 0 accordingly–starting *ex nihilo,* from nothing.

As scientists, we tend to use very sophisticated equipment, containing circuits that we barely understand and take for granted. Understanding and designing your own electronic circuits can allow you to control equipment on the microsecond scale. With sensors and switches, you can use electronics to build your own scientific equipment, and to record measurements. This course will introduce you to programming, electronics, data acquisition, system control strategies, and data analysis techniques.

The goal of this course is to introduce you to theoretical and applied concepts in electronic circuitry, for the purpose of collecting and analyzing experimental data. As the course was developed in the Leslie Dan Faculty of Pharmacy, many of the examples are rooted in pharmaceutics, the science involved in drug formulation design. The concepts discussed are nonetheless applicable to all quantitative research contexts where measurement and data collection are important. The course uses mixed teaching methods, with the first portion of each lecture as small-group didactic teaching, and the second portion as laboratory exercises to experiment with and illustrate the concepts covered. 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 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 complements learning activities.

With the recent advent of low-cost, consumer-level microprocessors (e.g. ATtiny, Arduino, Raspberry Pi, ESP8266, and new platforms continually being developed), affordable and accessible microprocessing has empowered researchers with resources to take experimental designs to new heights. Such microprocessors are relatively simple compared to the

complexities of today's computers; 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 instrument and computer. Serial communication ports were reliable only at slower speeds (e.g. 1200 bps). Serial communication was finicky, requiring 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 has evolved to support scientists and hobbyists alike. Knowledge of programming and circuitry will provide a solid foundation not only in experimental design and analysis for research, but in many other areas as well.

The development and commercialization of prototype boards such as Arduino, Raspberry Pi, and ESP8266 have empowered the electronics community with quicker and easier circuit prototyping. Due to economies of scale, electronic components have become very inexpensive over websites such as eBay, Alibaba, and Amazon. The electronics hobbyist demand has driven the development of modular electronic components marketed for general purposes, compatible with open-source platforms at +3.3V and +5V logic levels (e.g. opto-isolator power relay circuits, H-bridge motor controllers, and frequency-matched RF transmitters /receivers). Circuits that would previously need to be built from scratch 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.

Course Objectives

This course is broken up into eight sections, which loosely follow different "themes" in electronics. Learning objectives are defined at the end of every section to help guide your learning. Over-arching these learning objectives are the following course objectives.

After taking this course, you will …

1) Be able to properly interpret other people's circuit diagrams and draw your own diagrams in a manner that will enable you to accurately record and share your work with others. You will be able to draw circuit diagrams with enough detail so that others can build the same project. You will develop a foundation for "circuit diagram literacy". Many diagrams in this manual have been intentionally hand-drawn to reinforce that drawing a circuit diagram doesn't require circuit diagram-drawing software.

2) Have a working knowledge of some basic building blocks in electronics: resistors, capacitors, relays, transistors, MOSFETs, op-amps, voltage regulators, etc. This course will not expose you to all of the fundamentals (e.g. inductors, Zener diodes, transformers, and diode bridges have been intentionally omitted). However, the course provides enough background to start you off designing your own circuits, hopefully pointing you in a helpful direction. Whereas cook follows recipes, a chef creates them. Understanding how electronic building blocks fit together will enable you to go beyond following other people's circuit diagrams, synthesizing your own modules and ideas.

3) Be able to program in C++, which is a wonderful segue into learning other programming languages. Once you learn how to code in one language, learning another is often a matter of translation, which is much faster to pick up. You will learn the basics of writing a computer program, and have opportunities to write and compile your own code. You will learn proper programming etiquette, including commenting, selecting and working with appropriate variable types, writing subroutines, functions, and declaring local and global variables. After you are finished this course, you will be able to write, compile, and upload code for an Arduino microprocessor. These skills are transferrable to programming other microprocessors (e.g. the ATtiny85).

4) Be able to design, create, assemble, and test your own projects and scientific equipment, from raw sensor output to data filtering and recording.

5) Be able to develop and refine your troubleshooting skills, and become more confident in your circuit building and testing abilities. You will be empowered to solve your own problems, and recognize the value and complexity in the commercial electronics around you.

6) Be able to design and build circuits safely and carefully, always being mindful of the dangers of high voltage.

7) Have fun with electronics and programming!

The activities in this course have been carefully prepared assuming that you will work independently. There will be lots of opportunities to help troubleshoot each other's work; however, you are expected to perform these exercises on your own to help develop your skills at circuit designing and programming.

SECTION 1

INTRODUCTION TO ELECTRICITY

What You'll Be Learning	**Lecture:** Introduction to electricity and circuit diagrams. Ohm's Law, General Power Law, Power (AC *vs.* DC). Voltage, current, Resistance, and how to measure them. Electrical ground. Kirchhoff's Laws. Voltage and current dividers. Anode *vs.* cathode. Switches. Introduction to breadboarding.
What You'll Be Doing	**Activity 1-1:** 9V LED - resistor circuit. Measuring voltage, current, resistance. Calculating power. Adding a momentary switch. **Activity 1-2:** Set up a simple voltage divider circuit. Confirm the voltage divider equation by measuring the voltage difference across each resistor. **Demo:** Light Theremin
Files you will need	Not applicable.

What is Electricity?

Electricity, in the sense that we will be using it, can be described as the movement of electrons. Metals that have a portable, or *free* electron, can conduct electricity. Copper is one such metal, and consequently is commonly used to make wire. It has one "free" electron per atom. A *cloud* of these free electrons holds the metal together, forming metallic bonds. In any conductive metal, the free electrons move in Brownian-like random pathways. When there is a charge gradient, electrons generally move from an area of negative (–) charge to an area of positive (+) charge, or thinking about it differently, from a higher to lower concentration of negative charge.

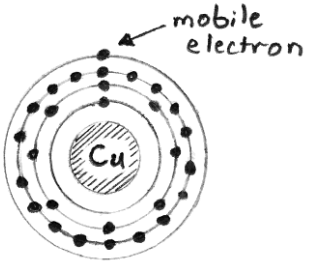

Figure 1-1. Mobile valence electron in outer shell of copper atom.

If electrons flow into one end of a copper wire, the outer-most electron leaves the orbit and flows to adjacent copper atoms, causing a chain of