

Welcome to Microcontroller Prototyping 180

In this course we will learn the fundamentals of building prototype electronics hardware using Arduino. We'll discuss the basics of programming in Arduino's C/C++ like language, as well as how to interface the Arduino with a variety of other devices. The goal of this course is to give a broad and well rounded introduction to prototyping. You won't leave this course with all the skills you need to complete a given project, it's a big wide world out there, but the skills you learn here will be used over and over again as you dive deeper – the fundamentals remain the same.

About your instructor

Kevin John has been a prototyping engineer and programmer with the Education and Public Outreach group at Sonoma State for several years. He graduated with a degree in physics from SSU in 2007 and has been at the group's lead programmer since that time. The group started transitioning into prototype hardware projects in 2011. Kevin has overseen the development of several Arduino based projects, as well as projects using other microcontrollers.

Rapid Prototyping Philosophy

This isn't a class on electrical engineering, and neither is it a class on computer science nor programming. The objective in this course is to teach only enough of either of those concepts to achieve our design goals. Because this is a course on rapid prototyping, we will adopt a philosophy of quickly developing a *working* prototype instead of spending lots of time working on a well *designed* prototype.

Materials

In order to avoid issues with parts, students are required to pay a materials fee to cover the costs of the hardware. To make the ordering process less error prone, we are not allowing students who may already own some of the parts to bring their own and defer the cost of parts. However, if you would like to work with a partner and share parts, two students may split a single parts kit. These parts are yours to keep. You may take them home between classes, or you may leave your parts at 180 studios, so nothing gets lost.

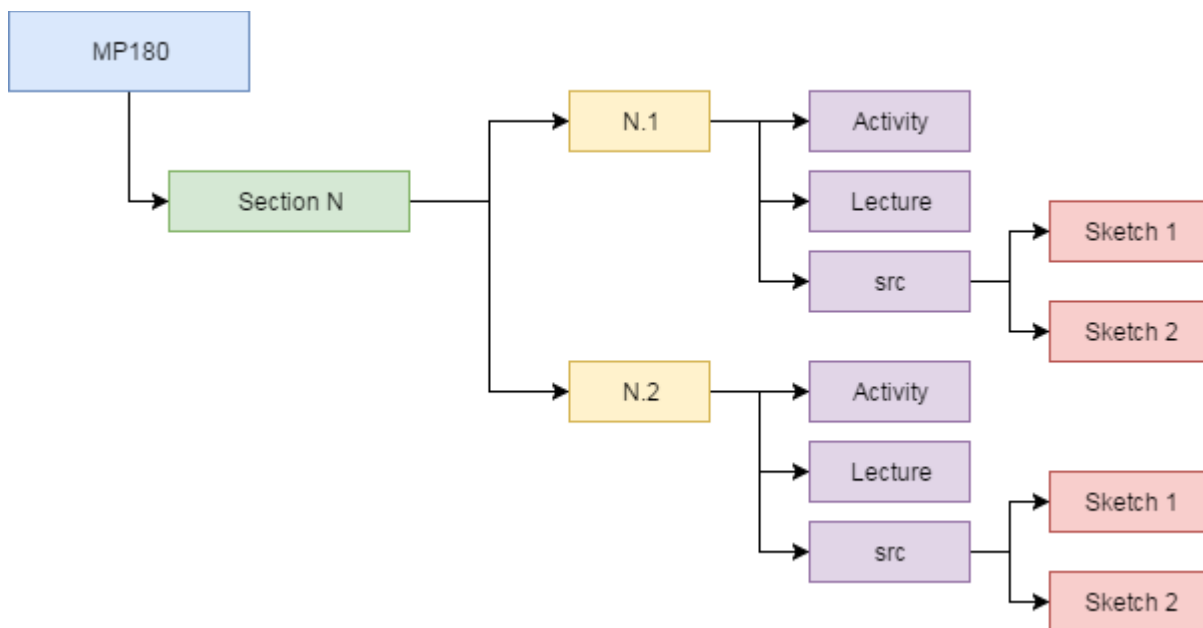
In order to keep the cost of parts low; we will re-use parts like the Arduino and the breadboards while building multiple prototypes. If you would like to be able to leave the prototypes you build assembled when you move on to the next prototype, and don't mind increasing the cost of your materials, you might benefit from adding two extra Arduinos and Breadboards to your equipment such that you never need to disassemble anything.

Class Schedule

This course has been designed to stretch over 4 two-hour class blocks. Since the pacing of the course is very fast, each class will be followed by an extra hour of workshop time, for a total of three hours. This is an opportunity to finish any tasks that run long; or even work on your own projects with help from your instructor. This course was not designed to allow students to pick which class sections are most interesting to them and only attend those. As each section builds on the previous, you should not expect to be able to miss a class and continue the next class session.

Folder Structure

Each class section has its own folder. Each folder has a number of sub-folders for each subsection. Each sub folder will contain two PDF files; an activity which will be worked through during the class session, and a lecture which may be read by the student at some point after the class session. You may choose not to read the lectures if you don't find them helpful or interesting, they are designed to augment the material we cover in the course and help with achieving a deeper understanding of the course material. For every section that contains code, which is most of them, you will also find a folder named src. In this folder, you sub folders containing Arduino sketches that you will load, modify, and run during the activities.



Course Tools:

Windows 10

The course will be taught using the Windows 10 computers that are already available at 180 Studios. This will ensure some uniformity, eliminate course fragmentation, and make things easier on your instructor. Of course, you may want to work on another operating system at home or on other projects. The course program files will run on any OS (Windows, Mac, Linux) without modification. Setting up your development environment on another machine isn't hard, and you can find instructions here: <https://www.arduino.cc/en/main/software>

Git

All the course files are available through our GIT repository.

<https://github.com/180Studios/mp180>

If you're not comfortable using GIT, you can download the files as a zip folder instead. Click on the green 'clone or download' button, and then click on the 'Download Zip' option.

There are two main branches of the code available. The 'student' branch will be loaded onto the computers during the course; this version has pieces of the code missing that are left as an exercise for the user to complete. The 'master' branch contains the completed versions of the source code as well as all the original versions of the course documents, source images, etc.