1/Introduction

Arduino is an open source *physical computing* platform for creating interactive objects that stand alone or collaborate with software on your computer. Arduino was designed for artists, designers, and others who want to incorporate physical computing into their designs without having to first become electrical engineers.

The Arduino hardware and software is open source. The open source philosophy fosters a community that shares its knowledge generously. This is great for beginners as help is often available geographically nearby and always online, at many different skill levels, and on a bewildering array of topics. Example projects are presented not just as pictures of the finished project, but include instructions for making your own or as a starting point for incorporation into your derivative or related projects.

The Arduino software, known as the Integrated Development Environment (IDE), is free. You can download it from *www.arduino.cc*. The Arduino IDE is based on the Processing language, which was developed to help artists create computer art without having to first become software engineers. The Arduino IDE can run on Windows, Macintosh, and Linux.

The Arduino board is inexpensive (about $30) and quite tolerant of common novice mistakes. If you do somehow manage to damage the main component on the Arduino Uno, it can be replaced for as little as $4.

The Arduino project was developed in an educational environment and is a very popular educational tool. The same open source philosophy that created the community which generously shares information, answers, and projects also shares teaching methods, curricula, and other information. Arduino has a special mailing list to facilitate discussion among anyone interested in teaching with or about Arduino.

Because the Arduino hardware and software are open source, you can download the Arduino hardware design and build your own, or use it as a starting point for your own project, based on (or incorporating) Arduino within its design, or simply to understand how Arduino works. You can do the same things with the software.

This book is designed to help beginners with no prior experience get started with Arduino.

Intended Audience

This book was written for the "original" Arduino users: designers and artists. Therefore, it tries to explain things in a way that might drive some engineers crazy. Actually, one of them called the introductory chapters of the first draft "fluff". That's precisely the point. Let's face it: most engineers aren't able to explain what they do to another engineer, let alone a regular human being. Let's now delve deep into the fluff.

This book is not meant to be a textbook for teaching electronics or programming, but you will learn something about electronics and programming while reading this book.

> After Arduino started to become popular, I realised how experimenters, hobbyists, and hackers of all sorts were starting to use it to create beautiful and crazy objects. I realised that you're all artists and designers in your own right, so this book is for you as well.
>
> —Massimo

Arduino builds upon the thesis work Hernando Barragan did on the Wiring platform while studying under Casey Reas and me (Massimo) at Interaction Design Institute Ivrea (IDII).

What Is Interaction Design?

Arduino was born to teach Interaction Design, a design discipline that puts prototyping at the centre of its methodology. There are many definitions of Interaction Design, but the one that we prefer is this:

Interaction Design is the design of any interactive experience.

In today's world, Interaction Design is concerned with the creation of meaningful experiences between us (humans) and objects. It is a good way to explore the creation of beautiful—and maybe even controversial—experiences between us and technology. Interaction Design encourages design through an iterative process based on prototypes of ever-increasing fidelity. This approach—also part of some types of conventional design—can be extended to include prototyping with technology; in particular, prototyping with electronics.

The specific field of Interaction Design involved with Arduino is physical computing (or Physical Interaction Design).

What Is Physical Computing?

Physical computing uses electronics to prototype new objects for designers and artists. It involves the design of interactive objects that can communicate with humans by using sensors and actuators controlled by a behaviour implemented as software running inside a microcontroller (a small computer on a single chip).

In the past, using electronics meant having to deal with engineers all the time, and building circuits one small component at a time; these issues kept creative people from playing around with the medium directly. Most of the tools were meant for engineers and required extensive knowledge.

In recent years, microcontrollers have become cheaper and easier to use. At the same time, computers have become faster and more powerful, allowing the creation of better (and easier) development tools.

The progress that we have made with Arduino is to bring these tools one step closer to the novice, allowing people to start

building stuff after only two or three days of a workshop. With Arduino, a designer or artist can get to know the basics of electronics and sensors very quickly and can start building prototypes with very little investment.

2/The Arduino Way

The Arduino philosophy is based on making designs rather than talking about them. It is a constant search for faster and more powerful ways to build better prototypes. We have explored many prototyping techniques and developed ways of thinking with our hands.

Classic engineering relies on a strict process for getting from A to B; the Arduino Way delights in the possibility of getting lost on the way and finding C instead.

This is the tinkering process that we are so fond of—playing with the medium in an open-ended way and finding the unexpected. In this search for ways to build better prototypes, we also selected a number of software packages that enable the process of constant manipulation of the software and hardware medium.

The next few sections present some philosophies, events, and pioneers that have inspired the Arduino Way.

Prototyping

Prototyping is at the heart of the Arduino Way: we make things and build objects that interact with other objects, people, and networks. We strive to find a simpler and faster way to prototype in the cheapest possible way.

A lot of beginners approaching electronics for the first time think that they have to learn how to build everything from scratch. This is a waste of energy: what you want is to be able to confirm that something's working very quickly so that you can motivate yourself to take the next step or maybe even motivate somebody else to give you a lot of cash to do it.

This is why we developed *opportunistic prototyping*: why spend time and energy building from scratch, a process that requires time and in-depth technical knowledge, when we can take