Contents

1 Understanding Arduino

This chapter is an introduction to the subject of Arduino. We see just what Arduino is, what can be done with it and the advantages it offers over competing platforms.

What is Arduino?

An Arduino is a small circuit board, which contains either an 8-bit or a 32-bit microcontroller, plus a handful of other components. Recent models, such as the Uno, also offer a USB interface, and a number of analog input pins as well as a number of digital input/output pins.

The concept behind the development of Arduino is to simplify the construction of interactive objects or environments and make them more accessible. To this end, it has been designed to be inexpensive and straightforward, thus providing a way for hobbyists, students, and professionals to create devices and projects that interact with their environment with sensors and actuators.

Typical examples of Arduino projects include simple robots, security systems and motion detectors. There are many more.

Arduino is about more than just hardware though. The microcontroller needs to be programmed and this introduces a software element in the form of an integrated development environment (IDE) that runs on personal computers. With it, users write programs (known as sketches) using the C or C++ programming languages.

The Arduino's microcontroller comes with a boot loader that makes it much easier to upload programs to the board's flash memory. The Arduino's competitors, in comparison, typically require the use of an external programmer. In keeping with the ethos behind Arduino, this keeps the programming side of things as straightforward as possible by allowing it to be done with a personal computer.

Because the connectors on Arduino boards are all standard types, it is possible to extend the basic Arduino by incorporating pre-built circuit boards, known as shields. Due to the design of the connector blocks, these can be stacked on top of one another, making it possible to create large and complex projects.

A number of Arduino boards are available. These include the Uno, the Duemilanove, the Diecimila, and the Mega. These are all designed for use with specific types of project and so come with differing specifications. Of all of them, the Uno is the most popular and can be used for the widest range of projects.

Arduino began in Italy at the Interaction Design Institute Ivera. This is a school of design education that focuses on interaction with digital devices and systems.

Arduino is the modern-day equivalent of those old electronic kits from yesteryear sold by companies such as Radio Shack and Heath

As a way of learning the basics of both electronics in general and programming, Arduino is an ideal tool.

Why Arduino?

The Arduino isn't the only low-cost computer on the market. There are a number of alternatives, all of which provide much the same capabilities. These include:

- Raspberry Pi
- CubieBoard
- Gooseberry
- APC Rock
- OLinuXino
- Hackberry A10

While we discuss Arduino in this book, it may be that one of the other low-cost computers is better suited to your requirements.

However, Arduino does have the following advantages:

Cost – Arduino boards are not as expensive as other microcontroller platforms. For many people, this is an important factor.

Cross-platform – Arduino software runs on Windows, Mac, and Linux operating systems. This is in contrast to most other microcontroller systems, which are limited to just Windows.

Simplicity & Flexibility – the Arduino programming environment is simple enough for beginners to quickly grasp, yet provides the flexibility that enables more complex projects to be tackled.

As an educational tool, it's conveniently based on the common open source Processing programming environment, so many people will already be familiar with it.

Open source software – Arduino software is open source and freely available for download from a number of websites. For those who don't have the ability to write their own programs or those who simply cannot be bothered, a huge range is available.

Open source hardware – the Arduino is based on the ATMEGA8 and ATMEGA168 microcontrollers, the plans for which are freely available under a Creative Commons license. Thus users can make their own version of the module to improve it or to increase its capabilities.

Arduino does offer advantages over the competition.

Which Arduino?

Arduino boards are not all the same – there are a number of types, each designed for different requirements. So before parting with the cash, it may pay you to take a look at what is available. Below, we look at the three most popular boards:

Arduino Uno

The most popular Arduino board of all is the Uno. This offers a range of features that make it a good all-purpose board.

It incorporates the ATmega328 chip as the controller and can be powered directly from USB, battery or an AC to DC adapter. The board operates at a voltage of 5 volts.

The board also offers 14 digital input/output pins, six of which can be used as pulse width modulation (PWM) outputs. In addition, it has six analog inputs, plus RX/TX (serial data) pins.

Memory provision is 32KB of flash memory, 2KB of static random access memory (SRAM), and 1KB of Electrically Erasable Programmable Read-Only Memory (EEPROM).

Variations of the Uno include:

The Leonardo – this is available both with and without headers, and offers a micro USB port.

The Ethernet – this comes with a RJ45 Ethernet socket rather than a USB port. It also offers a microSD card reader.

Pros of the Arduino Uno include:

- Simplicity
- Low cost
- Plug-and-play
- Plenty of resources available, e.g. tutorials, sample code, etc.
- Numerous extras available, e.g. shields and libraries

The Arduino Uno is the reference against which all other Arduino boards are compared.

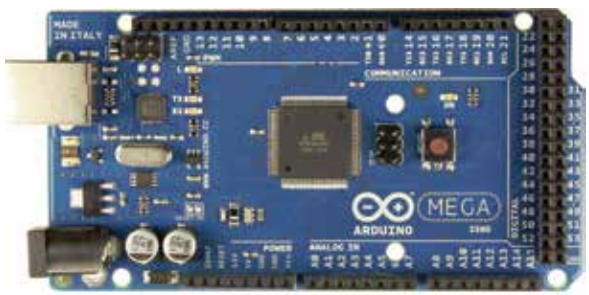
The Arduino Uno is the ideal board for beginners as it offers many options and features at a low price.

Cons include:

- Number of input/output pins is limited
- Small amount of memory provided can be restrictive
- 8-bit microcontroller

Arduino Mega 2560

This board is the Uno's big brother and is much the same, apart from being bigger. The increase in surface area allows it to offer 70 I/O pins (the Uno has just 14). Of these, 16 are analog inputs with the other 54 being digital. 15 of the digital pins can handle pulse width modulation (PWM). Also included are four RX/TX serial ports.



With regard to the controller, this is the ATmega2560 chip, which operates at 5 volts.

The Mega provides some four times more memory than the Uno. Specifically, 256KB of flash memory, 8KB of SRAM and 4KB of EEPROM.

The Mega is designed for more advanced users.

Variations of the Mega include:

The Due – this is based on a 32-bit ARM core microcontroller. Being much faster than the Uno, it is designed for use with more demanding applications. To this end, it is also furnished with more memory – 512KB of flash and 96KB of SRAM. Unlike other Arduino boards, the Due runs at 3.3 volts.

The ADK – this board is intended for use with Android cell phones.

If you are building a project for use with an Android cell phone, the Arduino ADK is the board to use.

Pros of the Arduino Mega 2560 include:

- Plenty of input and output pins
- Good amount of memory
- Plenty of resources available, e.g. tutorials, sample code, etc.
- Provides scope for bigger projects than does the Uno

…cont'd

Cons include:

- More expensive than the Uno (typically twice the price)
- Resources not as plentiful as for the Uno

Arduino Pro

As might be deduced from the name, the Arduino Pro is designed for professional use. The board is based on the ATmega168 or ATmega328 microcontrollers.

The Pro comes in both 3.3 volt and 5 volt versions. It has 14 digital input/output pins (six of which can be used as PWM outputs), and six analog inputs.

It also has holes for mounting a power jack, an ICSP header, and pin headers. A 6-pin header can be connected to an FTDI cable to provide USB power and communication to the board.

The Arduino Pro is designed for use in semi-permanent installations. The board doesn't provide pre-mounted headers thus allowing the user to employ whatever types of connectors are required by the project at hand.

Pros of the Arduino Pro include:

- Well suited for use in embedded projects
- Provides good flexibility when designing projects
- Soldered joints results in a higher level of reliability

Cons include:

- More expensive than the Uno
- Joints/connections need to be soldered

Of all the Arduino boards, the Pro offers the most options and flexibility.

Connections in the Arduino Pro need to be soldered by hand.

What Can You Do With It?

Now that you have some insight into what Arduino is, why you should use it in preference to competing platforms, and which version to use; you may want to consider just what can be done with it.

The first thing to be aware of is that because it comes with a microcontroller, an Arduino board can be used as the brains behind virtually any electronics project. This adds a huge amount of versatility over the simple electronic kits of yesteryear.

While you can do many things with a basic Arduino kit, complex projects will require extra parts.

By connecting a range of switches and sensors, Arduino can "sense" its surroundings; this includes light, sound, temperature, motion, pressure, etc. It can take this information and then with the aid of motors and other actuators, use it to interact with what's around it. As a random and very basic example, you could use Arduino to automatically control the lights in your house.

Devices that you can control with Arduino include switches, LEDs, motors, loudspeakers, GPS units, cameras, the Internet, and even smartphones and TVs.

Arduino can operate as an independent unit, it can be connected to a computer, or be connected to other Arduinos, or other electronic devices. In fact, with a bit of ingenuity pretty much anything can be connected to, and controlled by, Arduino.

Some Arduino projects are simply for fun. For example, a flashing LED cube, a tree climbing robot, a laser harp. Most, though, have a practical use. Typical examples are security systems, automated plant watering and smartphone garage door controllers. Arduino also provides a simple and inexpensive platform for building prototypes that enables ideas to be evaluated and turned into reality.

Arduino is used by artists, designers, and hobbyists.

However, there are applications for which Arduino is inherently unsuited. Because it has a limited amount of memory and slow processing capabilities, Arduino cannot be used for anything that requires serious processing power. This rules out video and audio processing, recording and output.

It also has a high power consumption, which means that battery powered applications can use up batteries at an alarming rate.

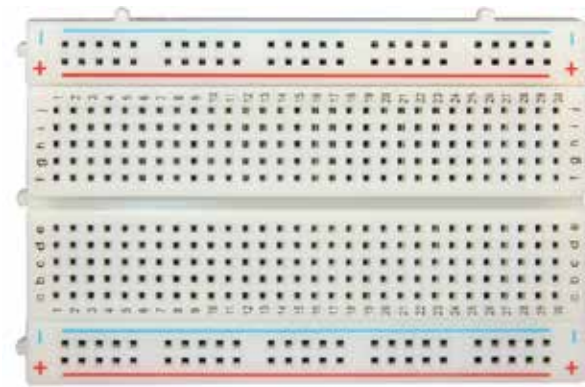
Basic Principles

Before we get down to the nitty gritty of Arduino, we'll take a look at the basics. This may help you decide whether or not Arduino is for you.

Hardware

An Arduino project is comprised of both hardware and software elements. We'll start with the former.

Breadboard – in your Arduino kit, there will be a blank board known as a breadboard, as shown below:



You will use this to build circuits. These are constructed by pushing the leads of electronic components such as resistors and capacitors into the breadboard's holes and then connecting them up with jumpers – all you need to know is which holes to push them into.

Note that there is no need to solder components in place – a big plus for those who have never soldered before or simply don't want the bother of it.

Breadboards offer two advantages: The first is that they enable circuits to be built quickly and easily. The second is that they make it very easy to rearrange components in order to correct mistakes or simply experiment.

Components – your Arduino kit includes a number of components, such as resistors, capacitors, switches, motors, etc. – see page 80 for a full list. These are the building blocks of your projects.

Soldering skills are not essential for Arduino.

Shields – the parts supplied in your kit are sufficient to build very basic projects. Once you get past these though, you will discover the need for more advanced circuits or, indeed, several of them, to build projects.

Arduino Ethernet Shield

If you have the ability there is nothing to stop you building these yourself with extension circuits. If you don't though, or simply don't want the bother, you can buy the required circuitry in the form of pre-built printed circuit boards. These are known as shields and they are readily available from a number of retail outlets.

If you don't have either the time or skills to build circuits yourself, you can buy ready-made shields.

The Microcontroller – having put together the various elements that comprise a project, you then need to give the Arduino the instructions necessary to make the project work, i.e. you need to program it.

The part of the Arduino board that interprets the instructions, or program, is the microcontroller.

However, before you can get started with programming, you will first need to set up the Arduino's software.

Software

This is not supplied with the board. You have to go to the Arduino website at **www.arduino.cc/en/main/software** and download it from there. The software is known as the integrated development environment (IDE).

There are IDE versions for Windows, Mac OS X, and Linux so you will be able to use Arduino with whatever operating system is on your PC.

Simply follow the prompts to install the software.

To program your projects you will need the use of a computer.

…cont'd

Programming – Having installed the IDE, you will now be able to write the necessary code for your project. This is done within the IDE on your computer, which provides a code editing window as shown below. The programs you write are known as sketches and can be manipulated in the same way as any other data, e.g. saved to the PC, uploaded to the Internet, etc.

With the sketch written, connect the Arduino board to the computer with the supplied USB cable and upload the code to the microcontroller. At this point your project is complete – all that remains is to see if it works.

Required Skills

The Arduino system is designed to be as straightforward as possible. However, that doesn't necessarily mean you can step straight in and get immediate results. Arduino projects involve more than just the Arduino itself. You will need other skills:

One is programming – all Arduino projects need to be programmed using the C and C++ programming languages. Another is electronics. All but the most simple of projects require external circuits to be constructed. This requires knowledge of electronic components, circuit design and construction.

Other skills you are likely to need include design, mechanics, fabrication and computing, to name just some. Obviously, this depends enormously on the scope of your projects.

C Programming in easy steps and C++ Programming in easy steps are available to learn how to program in these languages.

For those of you who don't have the requisite knowledge, the Internet is an enormous resource for all things to do with Arduino.