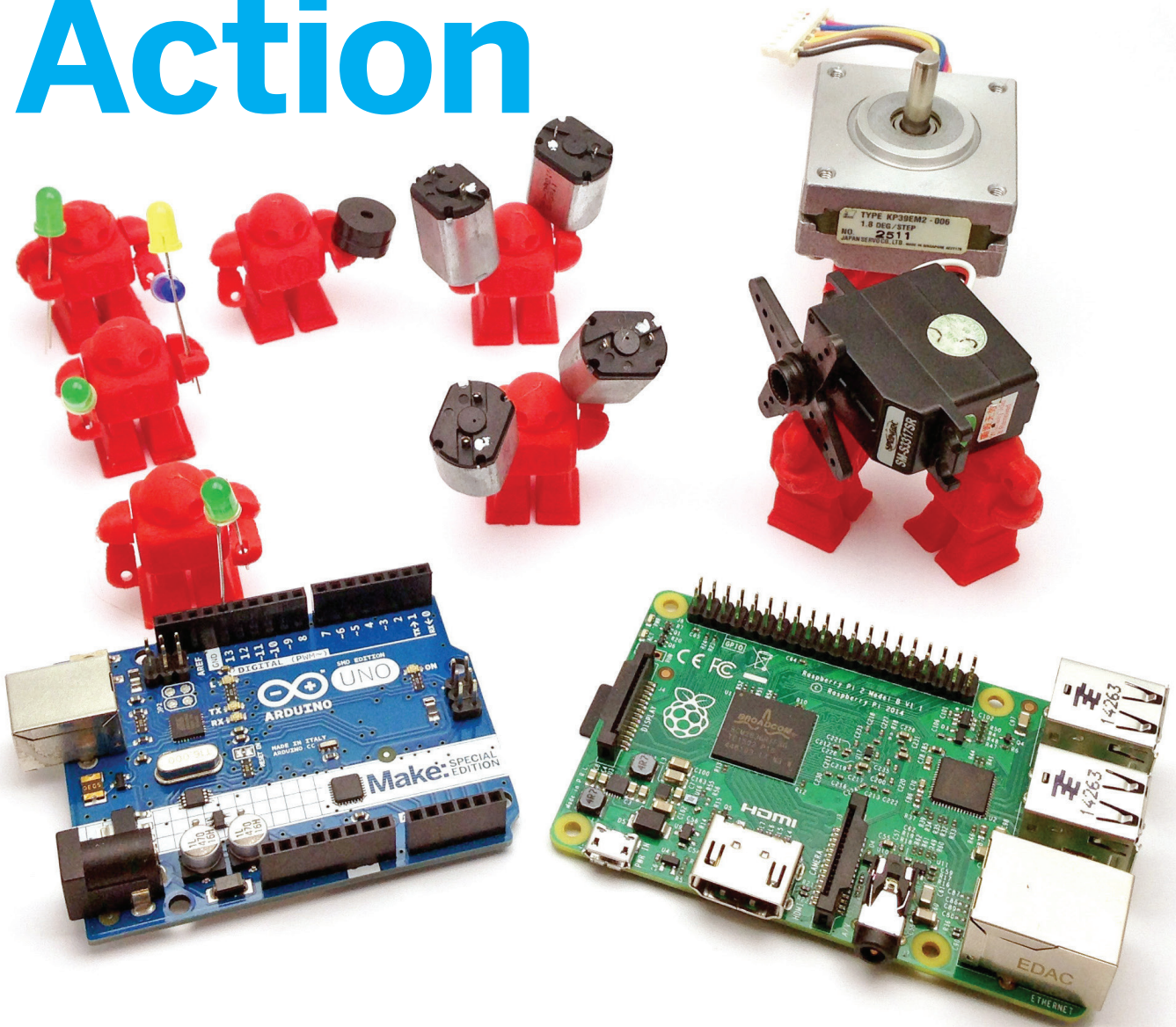


# Make: Action



Movement, Light, and Sound with  
Arduino and Raspberry Pi  
**Simon Monk**

# Make: Action

*Movement, Light, and Sound with  
Arduino and Raspberry Pi*

Simon Monk



## **Make: Action**

by Simon Monk

Copyright © 2016 Simon Monk. All rights reserved.

Printed in Canada.

Published by Maker Media, Inc., 1160 Battery Street East, Suite 125, San Francisco, CA 94111.

Maker Media books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (<http://safaribooksonline.com>). For more information, contact O'Reilly Media's institutional sales department: 800-998-9938 or [corporate@oreilly.com](mailto:corporate@oreilly.com).

**Editor:** Roger Stewart

**Production Editor:** Nicole Shelby

**Technical Reviewer:** Duncan Amos

**Copyeditor:** Jasmine Kwityn

**Proofreader:** Kim Cofer

**Indexer:** Ellen Troutman

**Interior Designer:** David Futato

**Cover Designer:** Karen Montgomery

**Illustrator:** Rebecca Demarest

January 2016: First Edition

### **Revision History for the First Edition**

2016-02-02: First Release

2016-07-29: Second Release

See <http://oreilly.com/catalog/errata.csp?isbn=9781457187797> for release details.

Make:, Maker Shed, and Maker Faire are registered trademarks of Maker Media, Inc. The Maker Media logo is a trademark of Maker Media, Inc. *Make: Action* and related trade dress are trademarks of Maker Media, Inc.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Maker Media, Inc. was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and authors assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

978-1-457-18779-7

[TC]

---

# Table of Contents

---

<b>1. Introduction</b> .....	<b>1</b>
Arduino and Pi .....	1
Raspberry Pi .....	1
Arduino .....	3
Choosing a Device: Arduino or Pi? .....	4
Alternatives .....	5
Summary .....	6
<b>2. Arduino</b> .....	<b>9</b>
What Is an Arduino? .....	9
Installing the Arduino IDE .....	11
Uploading a Sketch .....	13
The Book Code .....	14
Programming Guide .....	14
Setup and Loop .....	15
Variables .....	15
Digital Outputs .....	16
Digital Inputs .....	16
Analog Inputs .....	18
Analog Outputs .....	19

If/Else .....	20
Loops .....	21
Functions .....	22
Summary .....	23
<b>3. Raspberry Pi .....</b>	<b>25</b>
What Is a Raspberry Pi? .....	25
Setting Up Your Raspberry Pi .....	27
Preparing a MicroSD Card with NOOBS .....	28
Setting Up SSH .....	29
SSH on a Windows Computer .....	31
SSH on Mac or Linux .....	32
The Linux Command Line .....	32
The Book Code .....	34
Programming Guide .....	35
Hello, World .....	35
Tabs and Indents .....	35
Variables .....	36
if, while, etc. ....	36
The RPi.GPIO Library .....	37
The GPIO Header .....	37
Digital Outputs .....	38
Digital Inputs .....	39
Analog Outputs .....	39
Summary .....	39
<b>4. Quickstart .....</b>	<b>41</b>
Solderless Breadboard .....	41
How a Breadboard Works .....	43
Connecting a Breadboard to the Arduino .....	43
Connecting a Breadboard to the Raspberry Pi .....	44
Downloading the Software .....	45

Experiment: Controlling an LED .....	46
Parts List .....	46
Breadboard Layout .....	46
Arduino Connections .....	47
Arduino Software .....	48
Arduino Experimentation .....	49
Raspberry Pi Connections .....	49
Raspberry Pi Software .....	51
Raspberry Pi Experimentation .....	52
Comparing the Code .....	52
Experiment: Controlling a Motor .....	53
Parts List .....	53
Breadboard Layout .....	54
Experimenting Without Arduino or Raspberry Pi .....	55
Arduino Connections .....	56
Arduino Experimentation .....	57
Raspberry Pi Connections .....	57
Raspberry Pi Experimentation .....	58
Summary .....	58
<b>5. Basic Electronics .....</b>	<b>59</b>
Current, Voltage, and Resistance .....	59
Current .....	59
Voltage .....	60
Ground .....	61
Resistance .....	61
Power .....	62
Common Components .....	63
Resistors .....	63
Transistors .....	64
Diodes .....	71
LEDs .....	71

Capacitors .....	72
Integrated Circuits .....	72
The Ins and Outs of Connections .....	73
Digital Outputs .....	73
Digital Inputs .....	73
Analog Inputs .....	73
Analog Outputs .....	74
Serial Communication .....	74
Summary .....	74
<b>6. LEDs .....</b>	<b>75</b>
Regular LEDs .....	76
Current Limiting .....	76
Project: Traffic Signal .....	78
Parts List .....	79
Design .....	80
Arduino Connections .....	80
Arduino Software .....	81
Raspberry Pi Connections .....	82
Raspberry Pi Software .....	83
PWM and LEDs .....	84
RGB LEDs .....	85
Experiment: Mixing Colors .....	86
Hardware .....	86
Parts List .....	88
Arduino Connections .....	88
Arduino Software .....	89
Arduino Experimentation .....	90
Raspberry Pi Connections .....	90
Raspberry Pi Software .....	91
Raspberry Pi Experimentation .....	93
Summary .....	93

<b>7. Motors, Pumps, and Actuators</b> .....	<b>95</b>
Controlling Speed (PWM) .....	96
Experiment: Controlling the Speed of a DC Motor .....	96
Hardware .....	96
Arduino Connections .....	97
Arduino Software .....	98
Arduino Experimentation .....	100
Raspberry Pi Connections .....	100
Raspberry Pi Software .....	101
Raspberry Pi Experimentation .....	102
Controlling DC Motors with a Relay .....	102
Switching a Relay with Arduino or Raspberry Pi .....	103
Relay Modules .....	105
Experiment: Controlling a DC Motor with a Relay Module .....	106
Parts List .....	106
Wiring .....	106
Arduino Software .....	107
Raspberry Pi Software .....	108
Choosing a Motor .....	108
Torque .....	108
RPM .....	109
Gears .....	110
Gearmotors .....	110
Pumps .....	110
Peristaltic Pumps .....	111
Velocity Pumps .....	112
Project: Arduino House Plant Waterer .....	114
Design .....	114
Parts List .....	115
Construction .....	116
Software .....	118



Using the Project .....	120
Linear Actuators .....	120
Solenoids .....	122
Summary .....	123
<b>8. Advanced Motor Control .....</b>	<b>125</b>
H-Bridges .....	126
H-Bridge on a Chip .....	128
Experiment: Control the Direction and Speed of a Motor .....	130
Parts List .....	131
Design .....	132
Breadboard Layout .....	133
Experimenting .....	135
Arduino Connections .....	136
Arduino Software .....	137
Arduino Experimentation .....	139
Connecting the Raspberry Pi .....	140
Raspberry Pi Software .....	141
Raspberry Pi Experimentation .....	142
Other H-Bridge ICs .....	143
L298N .....	143
TB6612FNG .....	147
H-Bridge Modules .....	147
Project: Arduino Beverage Can Crusher .....	149
Parts List .....	149
Wiring .....	150
Mechanical Construction .....	151
Arduino Software .....	151
Summary .....	152
<b>9. Servomotors .....</b>	<b>153</b>
Servomotors .....	153

Controlling a Servo .....	154
Experiment: Control the Position of a Servomotor .....	155
Hardware .....	156
Parts List .....	156
Connecting the Arduino .....	157
Arduino Software .....	158
Experimenting with Arduino .....	159
Connecting the Raspberry Pi .....	160
Raspberry Pi Software .....	160
Experimenting with Raspberry Pi .....	162
Project: Pepe, the Dancing Raspberry Pi Puppet .....	162
Parts List .....	164
Design .....	164
Construction .....	165
Software .....	172
Using Pepe the Puppet .....	173
Summary .....	174
<b>10. Stepper Motors .....</b>	<b>175</b>
Stepper Motors .....	176
Bipolar Stepper Motors .....	176
Experiment: Controlling a Bipolar Stepper Motor .....	178
Parts List .....	179
Design .....	180
Arduino .....	181
Arduino Connections .....	181
Arduino Software (the Hard Way) .....	183
Arduino Software (the Easy Way) .....	185
Arduino Experimentation .....	186
Raspberry Pi .....	187
Raspberry Pi Connections .....	187
Raspberry Pi Software .....	188

Raspberry Pi Experimentation .....	190
Unipolar Stepper Motors .....	190
Darlington Arrays .....	191
Experiment: Controlling a Unipolar Stepper Motor .....	192
Hardware .....	193
Parts List .....	194
Arduino Connections .....	195
Raspberry Pi Connections .....	195
Software .....	196
Microstepping .....	196
Experiment: Microstepping on Raspberry Pi .....	197
Parts List .....	197
Raspberry Pi Connections .....	198
Software .....	198
Experimenting .....	200
Brushless DC Motors .....	201
Summary .....	202
<b>11. Heating and Cooling .....</b>	<b>203</b>
Resistive Heaters .....	203
Experiment: Resistor Heating .....	203
Parts List .....	203
Construction .....	204
Experimenting .....	204
Project: Arduino Random Balloon Popper .....	205
Parts List .....	206
Hardware .....	206
Software .....	208
Using the Balloon Popper .....	209
Heating Elements .....	209
Power and Energy .....	210
From Power to Temperature Increase .....	210

Boiling Water .....	210
Peltier Elements .....	211
How Peltier Elements Work .....	211
Practical Considerations .....	212
Project: Beverage Cooler .....	215
Parts List .....	215
Construction .....	216
Using the Project .....	217
Summary .....	218
<b>12. Control Loops .....</b>	<b>219</b>
The Simple Thermostat .....	219
Experiment: How Good Is On/Off Thermostatic Control? .....	220
Parts List .....	221
Design .....	222
Breadboard Layout .....	223
Software .....	224
Experimenting .....	227
Hysteresis .....	228
PID Control .....	229
Proportional (P) .....	230
Integral (I) .....	231
Derivative (D) .....	232
Tuning a PID Controller .....	232
Experiment: PID Thermostatic Control .....	233
Hardware .....	234
Arduino Software .....	234
Arduino Experimentation .....	236
Connecting the Raspberry Pi .....	240
Raspberry Pi Software .....	241
Raspberry Pi Experimentation .....	245
Project: A Thermostatic Beverage Cooler .....	246

Hardware .....	246
Parts List .....	246
Design .....	247
Construction .....	248
Arduino Software .....	250
Summary .....	254
<b>13. Controlling AC .....</b>	<b>255</b>
AC Switching in Theory .....	256
What Is Alternating Current? .....	256
Relays .....	257
Optoisolator .....	257
Zero-Crossing Optoisolators and Triacs .....	258
AC Switching in Practice .....	260
Relay Modules .....	260
Solid State Relays (SSRs) .....	262
The PowerSwitch Tail .....	263
Project: Raspberry Pi Timer Switch .....	263
Parts List .....	263
Construction .....	264
Software .....	265
Using the Project .....	266
Summary .....	266
<b>14. Displays .....</b>	<b>267</b>
LED Strips .....	267
Experiment: Controlling an RGB LED Strip Display .....	268
Parts List .....	269
Arduino Connections .....	269
Arduino Software .....	270
Raspberry Pi Connections .....	271
Raspberry Pi Software .....	273

I2C OLED Displays .....	275
Experiment: Using an I2C Display Module with Raspberry Pi .....	276
Parts List .....	276
Connections .....	277
Software .....	277
Experimentation .....	279
Project: Adding a Display to the Beverage Cooler Project .....	279
Parts List .....	280
Connections .....	280
Software .....	281
Summary .....	282
<b>15. Sound .....</b>	<b>283</b>
Experiment: Unamplified Speaker and Arduino .....	283
Parts List .....	284
Breadboard Layout .....	284
Arduino Software .....	285
Arduino Experimentation .....	286
Amplifiers .....	286
Experiment: Playing Sound Files on an Arduino .....	287
Parts List .....	287
Creating the Sound Data .....	287
Arduino Code .....	289
Arduino Experimentation .....	290
Connecting an Arduino to an Amplifier .....	290
Playing Sound Files on Raspberry Pi .....	293
Project: Pepe the Puppet Gets a Voice .....	293
Parts List .....	294
Breadboard Layout .....	295
Software .....	296
Using the Talking Puppet .....	298
Summary .....	298

<b>16. The Internet of Things</b> .....	<b>299</b>
Raspberry Pi and Bottle .....	300
Project: A Raspberry Pi Web Switch .....	301
Hardware .....	301
Software .....	302
Using the Web Switch .....	303
Arduino and Networks .....	303
Project: Puppet Twitter Notifier .....	305
Putting Pepe on the Internet .....	306
IFTTT (If This Then That) .....	309
Using the Project .....	311
Summary .....	312
<b>Appendix A. Parts</b> .....	<b>313</b>
<b>Appendix B. Raspberry Pi GPIO Pinout</b> .....	<b>319</b>
<b>Index</b> .....	<b>321</b>

# Introduction

# 1

The Arduino and Raspberry Pi make it easier than ever for a hobbyist to get into the world of electronics. Perhaps you want to set up a DIY home automation system so that you can control your lights and heating over your WiFi network, or simply control some motors.

This book will show you how to use the popular Raspberry Pi and Arduino platforms so that your Pi or Arduino can make and control movement, light, and sound.

## Arduino and Pi

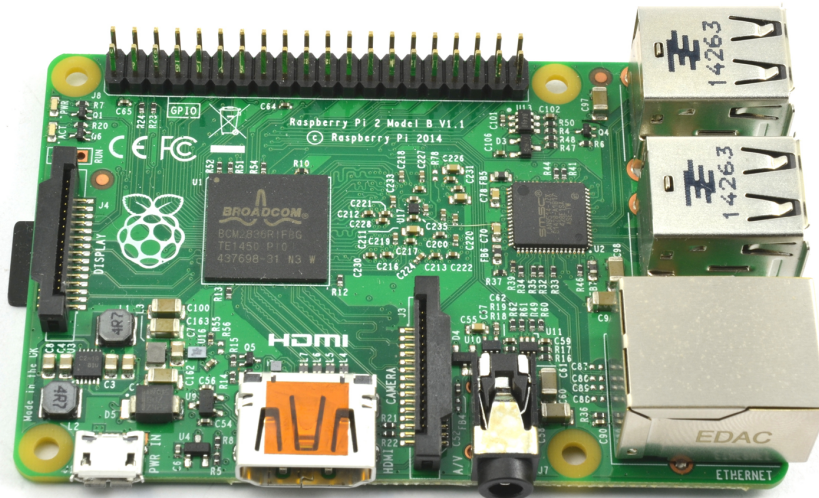
---

Although the Arduino and Raspberry Pi are both small, credit card-sized circuit boards, they are actually quite different devices. The Arduino is a very simple microcontroller board that does not run an operating system of any sort, whereas the Raspberry Pi is a tiny computer that runs Linux and also happens to be able to interface to external electronics.

## Raspberry Pi

If you are new to electronics, but comfortable using computers, then the Raspberry Pi is going to be the more familiar device. The Raspberry Pi (Figure 1-1) really is a very small version of a regular computer running Linux. It has USB ports for you to attach a keyboard and mouse as well as HDMI video output to connect to a monitor or TV and audio output.





**Figure 1-1** A Raspberry Pi 2

The Raspberry Pi has an Ethernet port to connect it to your network and will also accept USB WiFi adapters. Power is supplied to the Raspberry Pi using a microUSB socket.

A microSD card is used for storage rather than a conventional disk drive. This card contains both the operating system and all your documents and programs.

The Raspberry Pi was created in the UK, primarily to serve as a low-cost computer to help with teaching computer basics, particularly Python programming, to school kids. In fact, the name Pi is said to be derived from the *Py* of *Python*.

A few things set the Raspberry Pi apart from a regular desktop or laptop computer running Linux:

- It costs just \$40 (a cut-down Raspberry Pi called the model A+ is also available for a lower price and the model zero for an even lower price).
- It uses under 5 watts of power.
- The Raspberry Pi has a double row of general-purpose input/output (GPIO) pins that allow you to connect electronics directly to it (the pins can be seen in the upper-left part of [Figure 1-1](#)). From these pins, you can control LEDs, displays, motors, and all the different types of output devices that you will work with later in this book.

In addition, the Raspberry Pi can also be connected to the Internet using WiFi or a LAN cable, making it suitable for Internet of Things projects ([Chapter 16](#)).

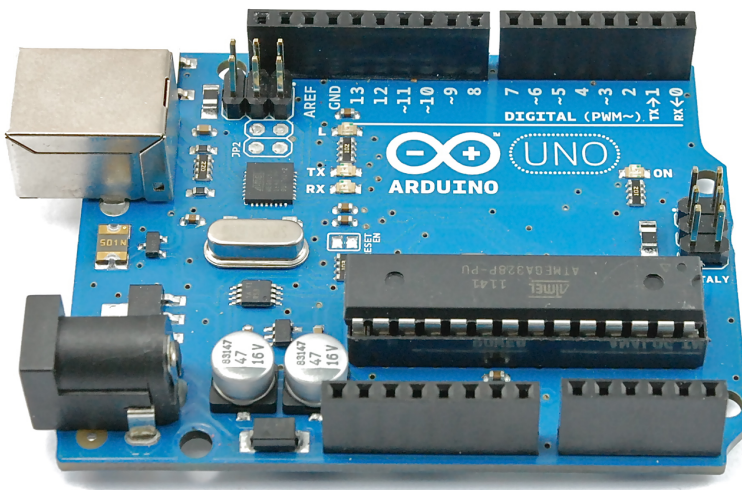
The specs for the Raspberry Pi 2 (the latest and best version at the time of writing) are as follows:

- ARM v7 900 MHz quad-core processor
- 1 GB DDR2 memory
- 10.100 BaseT Ethernet
- 4 USB 2.0 ports
- HDMI video out
- Camera interface socket
- 40-pin GPIO header (all pins operate at 3.3V)

If you are new to Raspberry Pi, there is a primer to get you up and running with the hardware and also the Python programming language in [Chapter 3](#).

## Arduino

There is quite a wide range of different Arduino models available. This book concentrates on the most widely used and popular Arduino model, the Arduino Uno ([Figure 1-2](#)). The Arduino is a little cheaper than a Raspberry Pi—you can purchase an Arduino Uno for around \$25.



**Figure 1-2** An Arduino Uno Revision 3

If you are accustomed to working on a regular computer, the Arduino's specifications will probably seem grossly inadequate. It has just 34 KB of memory (of various types). That means that the Raspberry Pi has roughly 30,000 times as much memory, without even

including the flash memory of the Pi's SD card! What's more, the Arduino Uno has a processor clocked at just 16MHz. You cannot attach a keyboard, mouse, or monitor to the Arduino, and it does not run an operating system.

You may wonder how this little device can actually do anything useful. The secret to the Arduino's usefulness lies in its very simplicity. There is no operating system to boot up, or other interfaces that you may not need in a project that would simply add cost and consume power.

Whereas the Raspberry Pi is a general computer, the Arduino concentrates on doing one thing well—connecting to and controlling electronics.

To program an Arduino, you need a regular computer (you can even use a Raspberry Pi if you want). On the computer you choose, you'll need to run an integrated development environment (IDE), which allows you to write a program to be downloaded to the built-in flash memory on the Arduino.

The Arduino can only run one program at a time, and once it is programmed, it will remember that program and automatically run it as soon as it is powered up.

Arduinos are designed to accept *shields*, which are boards that plug on top of the Arduino's input/output sockets and add extra hardware features, such as various types of display, as well as Ethernet and WiFi adapters.

You program an Arduino using the C programming language (you can find out more about programming and using an Arduino in [Chapter 2](#)).

## Choosing a Device: Arduino or Pi?

One of the reasons this book explains how to connect electronics to both Arduino and Raspberry Pi is that some projects are better suited to a Raspberry Pi and some to an Arduino. Other boards that lie in between these extremes are generally close enough to either an Arduino or Raspberry Pi that this book will be of use in working out how to use them.

When embarking on a new project, my rule of thumb is to use an Arduino by default. However, if the project has one of the following requirements, then a Raspberry Pi is probably a better choice:

- Internet or network connectivity
- The need for a large screen
- The need to attach keyboard and mouse
- The need for USB peripherals such as a web cam

With some effort and expense it's possible to expand the Arduino with shields to cover most of the preceding requirements. However, it will be harder to get everything working if you go this route, as none of these are native features of the Arduino in the same way as they are for the Pi.

Good reasons for using an Arduino over a Raspberry Pi include the following:

#### *Cost*

An Arduino Uno is cheaper than a Raspberry Pi 2.

#### *Startup time*

An Arduino does not need to wait while an operating system boots up. There is a small delay of about a second while it checks to see if a new program is being uploaded and then it's up and running.

#### *Reliability*

An Arduino is intrinsically a much simpler and tougher device than a Raspberry Pi and doesn't have the overhead of an operating system.

#### *Power consumption*

An Arduino uses about 1/10th of the power of a Raspberry Pi. If you need a battery- or solar-powered solution, then the Arduino is a better choice.

#### *GPIO output current*

A Raspberry Pi's GPIO pin should only be used to supply a maximum of around 16 mA. On the other hand, an Arduino pin is rated at 40 mA. So, in some cases, you can connect something (say a bright LED) to an Arduino directly, in a way that you couldn't with a Raspberry Pi.

The Arduino and the Raspberry Pi both are great devices to base a project on, and to some extent, the choice of which device to use will also be a matter of personal preference.

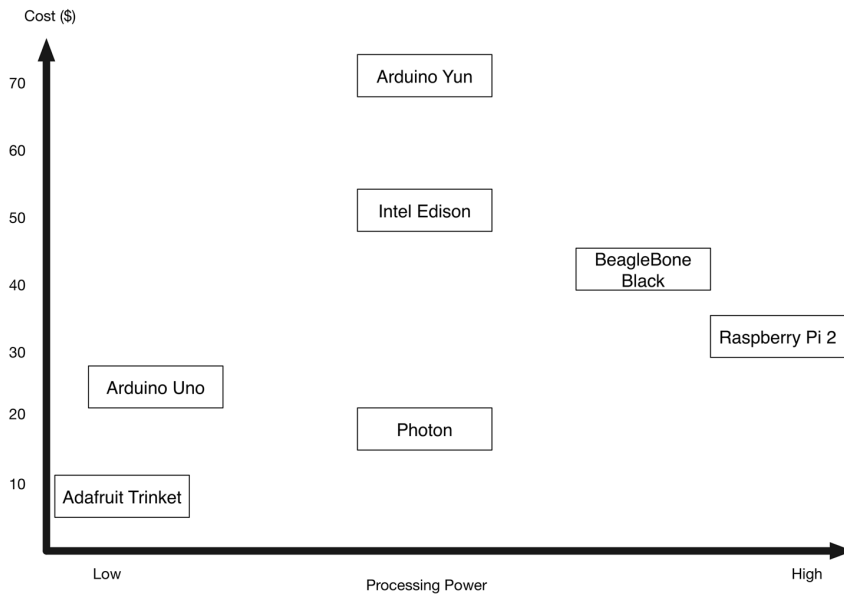
One important thing to remember when attaching external electronics to a Raspberry Pi is that it operates at 3.3V, rather than the 5V of Arduino. Connecting 5V to one of the Raspberry Pi's GPIO pins is likely to damage or destroy the GPIO pin or even the whole Raspberry Pi.

## **Alternatives**

The Arduino Uno and Raspberry Pi sit at each end of the range of devices that can be used to control things. As you might expect, the market has produced a whole host of other devices that sit in between these two extremes, in some cases trying to give the best of both worlds.

New devices are popping up all the time. The open source nature of Arduino has led to lots of variations on it, with designs for specific niches, such as controlling drones or interfacing to wireless sensors.

[Figure 1-3](#) shows a spread of some of the most popular devices in this area.



**Figure 1-3** *Embedded platforms*

Below the Arduino Uno, both in price and performance, is the Adafruit Trinket. This interesting board only has a few GPIO pins, but is otherwise fairly Arduino compatible. It's worth considering for a project that may just have one or two inputs or outputs.

There is a middle ground of products comprising the Arduino Yun, Intel Edison, and Photon that all have built-in WiFi capabilities and are intended for use with Internet of Things (IoT) projects (see [Chapter 16](#)). Of these, the Photon probably represents the best value. All three of these devices are programmed using Arduino C, so what you learn about using the Arduino will also apply to these boards.

The BeagleBone Black is very similar in concept to the Raspberry Pi. It too is a single-board computer and although the current BeagleBone Black version has fallen behind the Raspberry Pi 2 in terms of raw power, the BeagleBone Black does have some advantages over the Raspberry Pi. It has more GPIO pins and also has some pins that can act as analog inputs, a feature lacking in the Raspberry Pi 2. The BeagleBone Black can either be programmed in Python, in a similar way to the Raspberry Pi, or in JavaScript.

## Summary

This chapter provided a brief introduction to the Arduino and the Raspberry Pi. We discussed the advantages and disadvantages that each of these boards offer, and also looked at some alternatives. The next two chapters will get you started with using and programming the Arduino and then the Raspberry Pi.

If you have used Arduino and Raspberry Pi before, you might like to jump ahead to [Chapter 4](#) and use the Arduino and Raspberry Pi to Make some Action! You can always return to [Chapter 2](#) and [Chapter 3](#) if you need to.