
Designing the
Internet
of things
Adrian McEwen & Hakim Cassimally

Designing the Internet of Things

Adrian McEwen,
Hakim Cassimally

WILEY

This edition first published 2014

Registered office
John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom

A catalogue record for this book is available from the British Library.

ISBN 978-1-118-43062-0 (paperback); ISBN 978-1-118-43063-7 (ebook); 978-1-118-43065-1 (ebook)

Set in 9.5/11.5pt Minion Pro by Indianapolis Composition Services

Printed in the United Kingdom by TJ International

About the Authors

Adrian McEwen is a creative technologist and entrepreneur based in Liverpool. He has been connecting devices to the Internet since 1995—first cash registers, then mobile phones, and now bubble machines and lamps. He founded MCQN Ltd., an Internet of Things product agency and (along with Hakim and others) is co-founder of DoES Liverpool, a hybrid co-working/makerspace that incubates Internet of Things startups in NW England. He is also CTO of Good Night Lamp, a family of Internet-connected lamps. He was one of the first employees at STNC Ltd, which built the first web browser for mobile phones and was acquired by Microsoft in 1999. Adrian concentrates on how the Internet of Things intersects with people's lives and how heterogeneous networks of devices should work together, and lectures and speaks on these issues internationally. You can find him on the Internet at `www.mcqn.net` or follow him on Twitter as `@amcewen`.

For Jean, Les, and Christine, and in memory of Karen.

—Adrian

Despite an education in Italian and English literature, once **Hakim Cassimally** discovered software development, he hasn't looked back. He is a staunch proponent of Perl and was one of the organisers of YAPC::EU 2010 in Pisa. These days, however, he is likely to be writing Python for 3D printers or for civic hacking projects with mySociety.org. He co-founded (with Adrian and others) DoES Liverpool. His website is at `greenokapi.net`.

For my parents. This time I didn't start with the page numbers.

—Hakim

PUBLISHER'S ACKNOWLEDGEMENTS

Some of the people who helped bring this book to market include the following:

Editorial and Production
VP Consumer and Technology Publishing Director: Michelle Leete
Associate Director–Book Content Management: Martin Tribe
Associate Publisher: Chris Webb
Executive Commissioning Editor: Craig Smith
Project Editor: John Sleeva
Copy Editor: Chuck Hutchinson
Technical Editors: Daniel Soltis, Aaron Crane,
Alexandra Deschamps-Sonsino
Editorial Manager: Jodi Jensen
Senior Project Editor: Sara Shlaer
Editorial Assistant: Annie Sullivan

Marketing
Associate Marketing Director: Louise Breinholt
Marketing Manager: Lorna Mein
Senior Marketing Executive: Kate Parrett
Marketing Assistant: Polly Thomas

Composition Services
Compositor: Jennifer Mayberry
Proofreader: Susan Moritz
Indexer: Potomac Indexing, LLC

Acknowledgements

First off, we'd like to thank Craig Smith at Wiley for persuading us that capturing some of what we know into a book would be a good thing. Thanks also to the rest of the Wiley team and everyone who helped edit the book: John Sleeva, Daniel Soltis, Alexandra Deschamps-Sonsino, Chuck Hutchinson, and Aaron Crane. Despite our grumbling at the additional work of rewriting sections, the book is much better as a result.

We'd also like to thank everyone at DoES Liverpool for putting up with our sometimes distracted have-book-to-write periods in the run up to deadlines, and in particular our fellow organizers, John McKerrell, Andy Goodwin, Paul Freeman, Patrick Fenner, Steve Sparrow, and Ross Jones, who had to take up the slack in running the space.

—Adrian and Hakim

The acknowledgements section was one I never normally paid a lot of attention to, as it was the author thanking a load of people I didn't know. Having written a book, and realised how much help and support are given to the authors, I have a newfound appreciation for this section.

I'd like to thank Hakim for agreeing to join me in the adventure of actually doing the writing. Andy Huntington, Lawrence Archard, and Christopher Pett were all generous with their time and sharing their expertise in areas where I'm only in the early stages of exploring.

For endless encouragement and keeping me sane in the difficult bits, I need to thank my family and friends, especially Francis Irving, Andrew Dixon, Rob Stock, Neil Sowler, Kieran MacCourt, Jen Forakis, Colette MacCourt, Dan Lynch, and Neil Morrin.

And, finally, thanks to Bubblino for helping get me into this mess in the first place.

—Adrian

I tend to at least skim the acknowledgements and knew (in theory) that writing a book was a huge, collaborative effort. Experiencing it firsthand was awe-inspiring. We will almost certainly have forgotten to thank somebody: if that's you, we would love to make it up to you over a beverage of your choice.

Firstly, thanks to Adrian for suggesting we work together on the book; it's been an interesting project. Dave Cross, Kieren Diment, and Jodi Schneider gave invaluable early advice about publishing. Georgina Voss pointed me in the right direction on the ethics chapter, and Coen van Wyk kindly proof-read the first draft. Tomas Doran gave useful suggestions on asynchronous web programming for the online components chapter, and Aaron Crane stepped in as technical editor. Kelvin Lawson advised us on the current state of real-time operating systems, while Kiel Gilleade and Jen Allanson provided insights into physiological computing. Guy Dickinson gave his usual enthusiasm and support, and Jim Hughes has kept me regularly up-to-date with interesting ideas. Thanks, also, to Shaun Myhill at Electric Imp, Aaron at .:oomlout:., Antony McLoughlin, Ross Jones, and Francis Fish for donating or lending microcontrollers and books without which writing this book would have been impossible. My colleagues at mySociety have been exceptionally tolerant of my last-minute time off to finish the final drafts.

Having the opportunity to attend or speak at events such as OpenIoT, the IoT Howduino, TSB IoT round table, BBC's PlayIoT unconference, MadLab's IoT workshop, Internet World, LivLug, and SMC Liverpool helped shape the ideas of the book. Thanks to Alex Deschamps-Sonsino, Ed Borden, Usman Haque, Adam Greenfield, Hannah Goraya, Nick O'Leary, Ben Ward, Thomas Amberg, Charalampos Doukas, Stefan Ferber, Chris Adams, Gavin Sparks, Peter Bihr, Andrew Back, Laura James, Russell Davies, Rob van Kranenburg, Erik van der Zee, Martin Spindler, Matt Biddulph, Fiddian Warman, Rachel Jones, Alistair MacDonald, Natasha Carolan, Mark Holmes, Pat Link, Cefn Hoile, Thom Shannon, Alistair Houghton, Chris Holgate, Alastair Somerville, Matthew Hughes, Jessi Baker, Aine and Aidan McGuire, Grey Povey, Hwa Young Jung, Andy Piper, Sue Black, Travis Hodges, Bob Ham, Neil Morrin, Dan Lynch, Francis Irving, Conrad Mason, Pete Thomas. In addition, I have had great discussions online and offline with Javier Montaner, Paul Kinlan, Luke Petre, Paul Ede, and have received helpful comments and suggestions from Rob Nightingale, Zarino Zappia, Aldo Calpini, David Jones, Kirsty Sparrow, and Arto Nabito.

I worked with Greg McCarroll back in 2005. Though I didn't know it at the time, his connected train-aware alarm clock was the first Internet of Things object I ever came across. Greg was a gentleman as well as an innovator, and will be dearly missed.

Thanks, also, to Claire, Lolita, Rumplestiltskin, Sheri My, Wolfie, and Helvetica. Definitely.

—Hakim

Contents

Introduction

A COMPUTER PROCESSOR such as Intel's i486 used to cost around the same as a small car. Nowadays a chip with similar power is the price of a chocolate bar.

When processing power is so cheap, you can afford to put processors in places that you couldn't before—not just a business workstation or a home PC but also a telephone, an electricity meter, a bedside lamp, or a teddy bear. We can make objects intelligent. We can make them think and speak. Pundits have dubbed this "physical computing", "ubiquitous computing" or "ubicomp", or "the Internet of Things". Whatever you choose to call it, we are really talking about making magical things, enchanted objects.

In this book we look at the kinds of computer chips that can be embedded in objects ("microcontrollers" such as the Arduino) and take you through each step of the process from prototyping a Thing to manufacturing and selling it. We explore the platforms you can use to develop the hardware and software, discuss the design concepts that will make your products eye-catching and appealing, and show you ways to scale up from a single prototype to mass production.

WHAT THIS BOOK IS NOT

For starters, this book is not a specific guide to any given microcontroller. Although we look at the Arduino, Raspberry Pi, and other options, this detailed technical information will necessarily date more quickly than the rest of the material in the book. So we are more interested in showing the criteria for evaluating and choosing a platform.

Nor is this book a guide to particular cool projects to make. Rather, we survey some groundbreaking Things but mostly look at the general design principles that will, we hope, encourage you to make something fresh, beautiful, useful, and magical.

Finally, this isn't an academic treatise on the business infrastructure of tomorrow's Internet of Things—technologies such as 6LoWPAN and emerging M2M standards. We are far more interested in how to design, manufacture, and sell consumer-facing Things that will delight *people*.

WHO SHOULD READ THIS BOOK?

We certainly hope that this book, as a technical publication, will help software engineers, web developers, product designers, and electronics engineers start designing Internet of Things products. Indeed, we cover microcontrollers, electronics, embedded programming, and web APIs, among other technical topics of interest.

The book is also targeted at entrepreneurs, "makers" (designers, artists, craftspeople, and hobbyists), academics and educators, and anyone interested in getting an overview of this exciting upcoming technology. Even if you have little or no technical background in IT, much of the material is accessible to a general reader, and our coverage extends beyond the purely digital to topics in design, ethics, and business.

HOW TO USE THE BOOK

As we mentioned previously, this is not a "how-to" book, so you don't need any particular tools to read along. The more technical chapters do give some suggestions for microcontroller hardware or web development frameworks that you may choose to investigate further in parallel.

We have designed the book to flow from principles through prototyping to manufacture and business considerations, so you *can* certainly read it from cover to cover. However, you may prefer to take alternative paths through the book. Depending on your background and concerns, some chapters may be of greater interest, whereas others you may prefer to skip for now.

Part I, "Prototyping", introduces the Internet of Things and moves onto experimenting and creating your prototype project.

We recommend that all readers start with Chapter 1, "The Internet of Things: An Overview", which describes what the Internet of Things is and why it's happening now, and Chapter 2, "Design Principles for Connected Devices", where we set out a manifesto for consumer-facing Things for humans.

Chapter 3, "Internet Principles", is designed to be an accessible introduction which will be helpful to keep your ideas clear while you are building your Thing and thinking about how it communicates with the world. If you have a background in the Internet and web family of protocols, you can certainly skip this one.

The next chapters will be of most interest if you are planning to build a Thing yourself. To get a better understanding of the field, it is certainly worth reading Chapter 4, "Thinking About Prototyping", for general considerations about technology choices. For the engineer, maker, or technical person responsible for making a device, Chapter 5, "Prototyping Embedded Devices", applies the general principles to specific devices (at time of publication); Chapter 6, "Prototyping the Physical Design", discusses how to build the physical design of your prototype; and Chapter 7, "Prototyping Online Components", describes building the online components as web APIs.

Although many readers will have at least some general knowledge of programming, writing code for the small computers which often power connected devices has its own challenges. Chapter 8, "Techniques for Writing Embedded Code", presents some lessons learned in the trenches and will be useful to makers involved in projects with more complex computational requirements.

Part II, "From Prototype to Reality", moves beyond the world of making and prototyping and looks at what happens when your project meets the real world.

If you are an entrepreneur hoping to make money from your Internet of Things project, Chapter 9, "Business Models", examines business models which go far beyond simply selling devices. Of course, if you *are* planning to sell devices, then moving to manufacture presents a whole new set of problems, such as creating PCBs, sourcing materials, and getting certification, as we discuss in Chapter 10, "Moving to Manufacture".

Finally, technology always changes the world, and not always for the better. We began the book with design principles for making enchanted objects, and Chapter 11, "Ethics", discusses how ethical and moral principles are also essential if we want to keep the enchantment from going bad.

You can find more information about the book and the authors at `book.roomofthings.com` or by following `@aBookOfThings` on Twitter.

PRODUCTION NOTES

They say travel broadens the mind. It also helps you write. Parts of the book were written on a couple of flights, but much more on countless train journeys—mostly between Liverpool and London, but also elsewhere in the UK and in northern Italy. Former transport systems worked, too: the High Line in New York was an excellent venue for writing during an extended visit there early in the book's life.

The rest was written in and around Liverpool. In DoES Liverpool, in my flat by the cathedrals, holed up in Bold St. Coffee, or on the third floor of the majestic Central Library. When the weather permitted, even some down at the Pier Head, overlooking the river Mersey.

The main text was written in Vim in Markdown, on a Sony Vaio laptop running Ubuntu, and then edited in Microsoft Word on Windows on the same laptop.

—Adrian

Before even touching a keyboard, Adrian and I had several long, wide-ranging conversations about the Internet of Things in general and the content we wanted for the book. As I was quite new to the topic, this helped me form an understanding by getting to ask silly questions and challenge assumptions. The discussions helped us to reach a common "voice" for the book. We drafted a chapter together (which eventually became Chapter 4, "Thinking About Prototyping") and then split the remaining chapters evenly in accordance with our interests and knowledge. We have consistently reviewed each other's chapters before submitting them, which has helped maintain that shared voice through the whole process.

My drafts were written in Vim, converted from Markdown using Pandoc, and edited in LibreOffice, originally on an aging ThinkPad and then, when that finally gave up the ghost, on a MacBook Pro. Dropbox was invaluable for sharing the latest version of every document immediately. We wrote blog posts in Markdown, too, and published them with Jekyll.

—Hakim

PROTOTYPING

1

THE INTERNET OF THINGS: AN OVERVIEW

THE FIRST QUESTION that we should attempt to answer is, of course, what *is* the Internet of Things? Although the concepts we call on throughout this book are relatively straightforward, people have many different visions of what the phrase means, and many of the implications are hard to grasp. So we will take this question slowly in this chapter and look at it from a number of different angles.

What does the phrase "Internet of Things" mean? And how does it relate to the earlier buzzword "ubiquitous computing"? For those who are interested in the history of technological progress, where does the Internet of Things sit in the broad sweep of things, and why are we talking about it *now*? For those who understand best through metaphors, we look at the idea of *enchanted objects*, an image which has described technology for millennia but which is especially potent when describing the Internet of Things. For the more practical readers who understand by seeing examples of real things, we sketch out some of the exciting projects that give a good flavour of this exciting field. Let's start with this last approach, with a short piece of "design fiction".

THE FLAVOUR OF THE INTERNET OF THINGS

The alarm rings. As you open your eyes blearily, you see that it's five minutes later than your usual wake-up time. The clock has checked the train times online, and your train must be delayed, so it lets you sleep in a little longer. (See http://makezine.com/magazine/make-11/my-train-schedule-alarm-clock/.)

In your kitchen, a blinking light reminds you it's time to take your tablets. If you forget, the medicine bottle cap goes online and emails your doctor to let her know. (See www.vitality.net/glowcaps.html.)

On your way out of the house, you catch a glow in the corner of your eye. Your umbrella handle is lit up, which means that it has checked the BBC weather reports and predicts rain. You sigh and pick it up. (See www.materious.com/#/projects/forecast/.)

As you pass the bus stop on the way to the station, you notice the large LCD display flash that the number 23 is due. It arrives when you turn the next corner. When the bus company first installed those displays, they ran on the expected timetable information only, but now that every bus has GPS tracking its location, they simply connect to the bus company's online service and always give the updated information. Various transport organizations have implemented this. London's TfL has some useful information on their signs at www.tfl.gov.uk/corporate/projectsandschemes/11560.aspx.

When you get to the station, your phone checks you in automatically to a location-based service (such as Foursquare). On your mantelpiece at home, an ornament with a dial notices the change and starts to turn so that the text on it points to the word "Travelling". Your family will also see later that you've arrived at "Work" safely. (See http://wheredial.com.)

On your lunch break, a pedometer in your training shoes and a heart monitor in your wrist band help track your run around the block. The wrist band's large display also makes it easy to glance down and see how fast you are running and how many calories you've burned. All the data is automatically uploaded to your sports tracking site, which also integrates with your online supermarket shopping account to make it easy to compare with how many calories you've eaten. (See http://nikeplus.nike.com/plus/.)

As you can see from the preceding links, each of these products is feasible with today's technology. Each has been prototyped, and many of them exist as craft or mass-market products.

THE "INTERNET" OF "THINGS"

We've looked at a number of examples of the Internet of Things, so what is the common thread that binds them together? And why the name? All the cases we saw used the *Internet* to send, receive, or communicate information. And in each case, the gadget that was connected to the Internet wasn't a computer, tablet, or mobile phone but an object, a *Thing*. These Things are designed for a purpose: the umbrella has a retractable canopy and a handle to hold it. A bus display has to be readable to public transport users, including the elderly and partially sighted and be able to survive poor weather conditions and the risk of vandalism. The sports bracelet is easy to wear while running, has a display that is large enough and bright enough to read even when you are moving, and will survive heat, cold, sweat, and rain.

Many of the use cases could be fulfilled, and often are, by general-purpose computers. Although we don't carry a desktop PC around with us, many people do carry a laptop or tablet. More to the point, in almost every country now, most people do carry a mobile phone, and in many cases this is a smartphone that easily has enough power for any task one could throw at a computer. Let's see how well one could replicate these tasks with a smartphone.

Viewing your bus provider's timetable with a smartphone web browser seems to fulfil the same function at first glance. But just consider that last phrase, "at first glance". On arriving at the bus stop, one can simply glance at the computerised timetable and see when the next bus is due. With a smartphone, if you have one and can afford the data use (which may be prohibitive if you are a foreign tourist), you have to take the phone out of your pocket or bag, unlock it, navigate to the right website (this may be the slowest and most complicated part of the process, whether you have to type the URL or use a QR code), and read the data from a small screen. In this time, you are not able to fully concentrate on the arriving buses and might even miss yours.

You can track your runs with an app on your smartphone, and many people do: the phone has GPS, many other useful sensors, processing power, an Internet connection, and a great screen. But it turns out that such a phone isn't easy to carry on a run without worrying about dropping it or getting it wet. Plenty of carrying options are available, from a waist bag to an arm strap. The latter, in theory, enables you to read the device while you are running, but in practice reading details on the screen can be hard while you are jiggling up and down! To get around this difficulty, apps such as RunKeeper provide regular audio summaries which can be useful (`www.runkeeper.com`). Ultimately, a phone is a perfectly capable device for

tracking your run, and most runners will find it a sufficient, comfortable, and fun way of logging their running data. However, others may well prefer a device worn as a watch or wristband, designed to be read on the move, worn in the rain, and connected to peripherals such as heart monitors.

Of course, no mobile phone (or even tablet or laptop) is large enough or waterproof enough to use as an umbrella. However, you could pair a smartphone with a normal "dumb" umbrella, by checking an app to see whether it is likely to rain later, before you leave the house. Unlike a calm, subtle light in the umbrella stand, glimpsed from the corner of your eye as an ambient piece of information to process subconsciously when you pass it on the way out of your home, an app requires you to perform several actions. If you are able to establish and maintain the habit of doing this check, it will be just as effective. Rather than having greater capabilities, the smart umbrella simply moves the same intelligence into your environment so that you don't have to change your routine.

So the idea of the Internet of Things suggests that rather than having a small number of very powerful computing devices in your life (laptop, tablet, phone, music player), you might have a large number of devices which are perhaps less powerful (umbrella, bracelet, mirror, fridge, shoes). An earlier buzzword for roughly the same concept was "ubiquitous computing", also known by the ugly portmanteau "ubicomp", and this also reflects the huge number of possible objects that might contain computing technology. Now that the Internet is a central pipe for data, it's hard to imagine, for example, a PC that doesn't have an always-on broadband connection. Younger readers may never have seen such a thing. As technologist and columnist Russell Davies joked at the 2012 Open Internet of Things Assembly in London:

> *I can't understand why teddy bears did not have wifi before. A bear without wifi is barely alive, a semi-bear.*
>
> —http://storify.com/PepeBorras/opent-iot-assembly

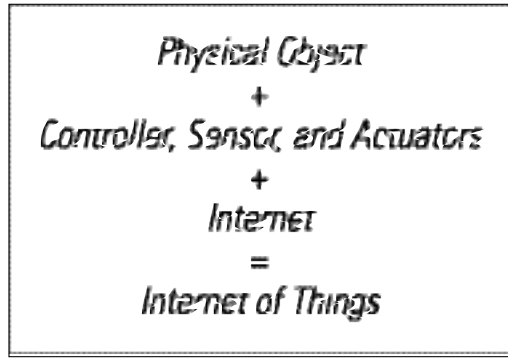
The definition of ubicomp, however, would also include the Glade air fresheners which release scent when they detect movement in the room as part of its domain. That is to say, such a device is an intelligently programmed computer processor, driven by sensors in the real world, and driving output in the real world, all embedded into an everyday object. These factors make this ubicomp, and it is only differentiated from the "Internet of Things" by the fact that these days most of the really interesting things done with computing also involve an Internet connection.

But what does it mean to "connect an object to the Internet"? Clearly, sticking an Ethernet socket into a chair or a 3G modem into a sewing machine doesn't suddenly imbue the object with mysterious properties. Rather, there has to be some flow of information which connects the defining characteristics of the Thing with the world of data and processing represented by the Internet.

The Thing is present, physically in the real world, in your home, your work, your car, or worn around your body. This means that it can receive inputs from your world and transform those into data which is sent onto the Internet for collection and processing. So your chair might collect information about how often you sit on it and for how long, while the sewing machine reports how much thread it has left and how many stitches it has sewn. In subsequent chapters, we talk a lot about "sensors".

The presence of the Thing also means that it can produce outputs into your world with what we call "actuators". Some of these outputs could be triggered by data that has been collected and processed on the Internet. So your chair might vibrate to tell you that you have received email.

We could summarize these components in the following appealingly simple (though, of course, also simplistic) equation:



An equation for the Internet of Things.

Note that in all the cases we've looked at, the form of the object follows the function of the Thing: your chair is designed to sit on, the sewing machine to sew at, and so on. The fact of also being connected to the Internet and having general-purpose computing capabilities doesn't necessarily have an impact on the form of the object at all. (One might argue that current-generation smartphones and tablets are in forms optimized for use as general-purpose computers, not as portable telephony devices. Certainly, on seeing the number of phones with scratched screens, one could ask whether they are designed to be easy to hold securely and resistant to drops and the impacts of everyday use.)

THE TECHNOLOGY OF THE INTERNET OF THINGS

In starting to define the Internet of Things, we compared it to the earlier concept of ubiquitous computing. We could compare that, in turn, with Bill Gates's famous vision in 1977 of "a computer on every desk and in every home" (`http://danbricklin.com/log/billg_entwof.htm`) and again with the earlier notion of a computer as an astonishingly expensive and specialised machine, accessible only to universities, some forward-thinking global corporations, and the military. It is worth taking a little time to look at the Internet of Things through a lens of the history of technology to more clearly understand how and where it fits.

Technology's great drivers have initially been fundamental needs, such as food and water, warmth, safety, and health. Hunting and foraging, fire, building and fortifications, and medicine grow out of these needs. Then, because resources for these things are not always distributed where and when one might like, technological advances progress with enabling and controlling the movement of people, their possessions, livestock, and other resources. Trade develops as a movement of goods from a place where they are plentiful and cheap to one where they are rare and valuable. Storage is a form of movement in time—for example, from harvest time, when food is plentiful and cheap, to the following winter, when it is highly valued.

Information becomes key, too—hence, the development of language to communicate technology to others. Travellers might pass on messages as well as goods and services, and an oral tradition allows this information to pass through time as well as space. The invention of writing makes this communication ever more important and allows, to some extent, human lives to be preserved in words by and about writers, from the ancient philosophers and poets to the present day. From writing, via the telegraph, radio, and television, to digital information, more and more technology has been about enabling the movement of information or doing interesting things with that information.

But the other human needs we looked at haven't ceased to exist, nor will they. We still need to eat and drink. We still need light and warmth. We still need love and friendship. We still need chairs, clothes, and shoes; means of transport and communication; and ways to entertain ourselves. The shape and details of all of these things will change but not the needs they address.

As technology has progressed, new categories of objects have been created: in the electronic age, they have included telephones, radios, televisions,

computers, and smartphones. As with most new technology, these devices tended to start out very expensive and gradually come down in price. Demand drives down prices, and research leads to optimization and miniaturisation. Ultimately, it becomes not just possible but also feasible to include functionality that would previously have required its own dedicated device *inside* another one. So although a television screen would originally have physically dominated a living room, not only are today's flat-screen panels more compact, but the technology is so ubiquitous that a high-resolution screen capable of displaying television content can be embedded into a door frame or a kitchen unit, and of course, even smaller screens can find their way into music players and mobile phones.

Similarly with computers, it has become so cheap to produce a general-purpose microchip in devices that your washing machine may contain a computer running Linux, the cash register at the supermarket may run on Windows, and your video player may run a version of Apple's OS X. But as we've already hinted at, mere computing power isn't a sufficient precondition for the Internet of Things. Rather, we are looking at computing power linked on the one hand to electronic sensors and actuators which interact with the real world and on the other to the Internet. It turns out that the rapid sharing and processing of *information* with services or other consumers is a huge differentiator.

As an example, let's consider the computers that exist in modern cars: they have myriad sensors to determine how well the car is running—from oil gauge and tyre pressure to the internals of your engine. As well as diagnostics, computerized brakes may assist the driver when the processor spots conditions such as the wheels locking or spinning out of control. All this is local information, and although the processing and analysis of this data may be highly sophisticated, it will be limited to whatever your car manufacturer has programmed. But perhaps your car also tracks your location using GPS: this is external (although not necessarily Internet-related) data. High-end cars may communicate the location back to a tracking service for insurance and anti-theft purposes. At this point, the car carries computing equipment that is able to not just passively consume data but also to have a dialogue with an external service. When your car's computer is connected to the Internet (regularly or permanently), it enables services such as responding to traffic conditions in real time by rerouting around them. Your GPS might already supply such data, but now it can be created in real time by "social route planning" based on the data aggregated from what other connected drivers nearby are doing. When the previously internal data gets connected to the Internet, the ways it can be processed, analysed, aggregated, and remixed with other data open up all the possibilities that we've seen in existing connected areas and indeed new ones that we can't yet imagine.

So there is a real change to an object or appliance when you embed computing power into it and another real change when you connect that power to the Internet. It is worth looking at why this latter change is happening *now*.

When the Internet moved out of academia and the military, with the first commercial Internet service providers (ISPs) opening for business in the late 1980s, the early adopters of the consumer Internet may have first gone online with a computer running an Intel 486 chip, costing around £1500, or around the price of a small car. Today a microchip with equivalent power might set you back around £0.50, or the price of a chocolate bar. The rapid rise of processing power, and the consequent cost decrease, is not a new insight: it is widely known as Moore's law (the rule of thumb, suggested by the co-founder of Intel, that says the number of transistors you can fit on a silicon chip will double every 18 months).

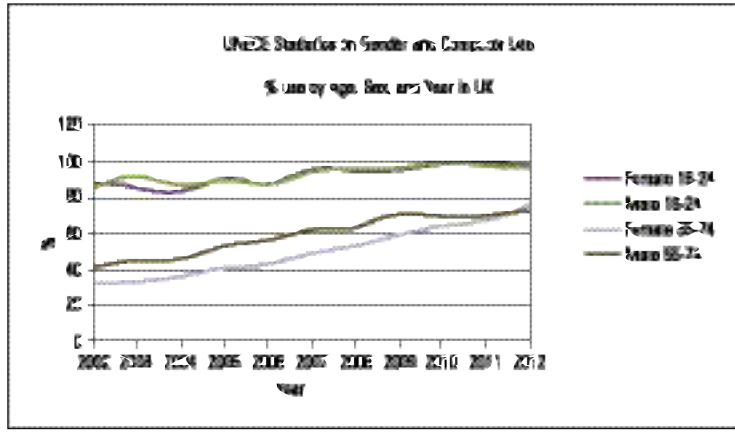
However, the kind of price difference we've mentioned isn't merely a question of degree: it is a *qualitative* as well as a quantitative change. This is a "long tail" phenomenon through which we have now hit the right price/performance sweet spot that means the cost of the computing horsepower required to talk to the Internet has fallen to a level where adding a network or computing capability is akin to choosing what type of material or finish to use—for example, whether to use a slightly more expensive wood veneer. Either option would add a little to the cost of the product but could also add disproportionately to its value to the customer. When Internet-capable computing cost thousands of pounds, this wasn't an option, but now that it costs tens of pence, it is.

So the price of computing power has come down to affordable levels, but this is only part of the story. Manufacturers of electronic products have started to incorporate general-purpose computer CPUs into their products, from washing machines to cars, as they have seen that it has become, in many cases, cheaper to do this than to create custom chips. The wealth of programming and debugging resources available for these platforms has made them attractive to hobbyists and the prototyping market, leading to the proliferation of the microcontrollers, which we look at in Chapter 4, "Thinking About Prototyping", and Chapter 5, "Prototyping for Specific Devices".

Internet connectivity is also cheaper and more convenient than it used to be. Whereas in the past, we were tied to expensive and slow dial-up connections, nowadays in the UK, 76% of adults have broadband subscriptions, providing always-on connectivity to the Net. Wired Ethernet provides a fairly plug-and-play networking experience, but most home routers today also offer WiFi, which removes the need for running cables everywhere.

While having an Internet-accessible computer in a fixed location was useful to those who needed to use it for work or studies, it would often be monopolized disproportionately by male and younger members of the family for general browsing or gaming. Now that the whole family can go online in the comfort of the living room sofa or their own room, they tend to do so in greater numbers and with ever greater confidence.

> *We hope the reader will excuse the preceding generalisation. As shown in the following figure, computer use in the UK between genders for the 16–24 age group is near identical since 2002. For the 55–74 group, there is a clear gap which persists, despite increasing take-up for both genders, until a tipping point around 2010 (http://w3.unece.org/pxweb/database/STAT/30-GE/09-Science_ICT/). Our hypothesis is that the shift is due, at least in part, to processing power and connectivity becoming cheap, widely available, and convenient. Not entirely coincidentally, these are the same factors we suggest help give rise to the Internet of Things.*



UNECE statistics on gender and computer use.

For situations in which a fixed network connection isn't readily available, mobile phone connectivity is widespread. Because the demand for connectivity is so great now, even embryonic solutions such as the whitespace network are available to use the airspace from the old analogue TV networks to fill gaps.

Another factor at play is the maturity of online platforms. Whereas early web apps were designed to be used only from a web browser, the much heralded "Web 2.0", as well as bringing us "rich web apps", popularized a style of

programming using an Application Programming Interface (API), which allows other programs, rather than just users, to interact with and use the services on offer. This provides a ready ecosystem for other websites to "mash up" a number of services into something new, enables mobile phone "Apps", and now makes it easy for connected devices to consume.

As the online services mature, so too do the tools used to build and scale them. Web services frameworks such as Python and Django or Ruby on Rails allow easy prototyping of the online component. Similarly, cloud services such as Amazon Web Services mean that such solutions can scale easily with use as they become more popular. In Chapter 7, "Prototyping Online Components", we look at web programming for the Internet of Things.

ENCHANTED OBJECTS

The best known of Arthur C. Clarke's "three laws of prediction" states

> *Any sufficiently advanced technology is indistinguishable from magic.*
> —http://en.wikipedia.org/wiki/Clarke's_three_laws

We've already seen how technology has evolved to meet our needs and desires. The parallel invention of magic serves largely similar goals. After all, the objects in folktales and fairy tales are often wish-fulfilment fantasies to fill the deepest desires: if only I had enough to eat; if only my mother was well again; if only I could talk to my friend even though I'm far away; if only I could get home; if only I didn't have to work every hour of the day to earn enough money for my family to eat. Literary and anthropological scholars have long studied fairy tales for the lessons that can be learnt about the basic rules of human narrative and meaning and have analysed the characters, storylines, and objects found within them. For example, the formalist scholar Vladimir Propp categorized the folktales of his native Russia and categorised their plot elements into 31 functions, including "violation of interdiction", "villainy", "receipt of a magical agent", "difficult task", and so on.

More recently, and from the point of view of a Silicon Valley entrepreneur and technologist, David Rose has talked about Enchanted Objects at TEDx Berkeley (`http://tedxtalks.ted.com/video/TEDxBerkeley-David-Rose-Enchant`) and has categorised various objects drawn from fairy tales and fantasy literature in ways that apply as much to technological objects. For *Protection*, just as magical swords and helmets protected the