Raspberry Pi Home Automation with Arduino
Second Edition

Revolutionize the way you interact with your home and become part of the rapidly growing group of hobbyists and enthusiasts by combining the powerful Raspberry Pi with an Arduino board.

Starting with an introduction to the Raspberry Pi and Arduino boards you will set up your very own home automation system through a series of exciting projects, making your life a lot easier.

Build a thermostat, a Raspberry Pi control device, a parcel delivery detector, and a water and damp detector, and then move on to more complex projects including a motor mechanism to control your curtains.

By the end of this practical guide, you will be a confident user who can program, set up, and run a wide range of applications using both the Raspberry Pi and the Arduino.

Who this book is written for
If you are new to the Raspberry Pi, the Arduino, or home automation and wish to develop some amazing projects using these tools, then this book is for you. Any experience in using the Raspberry Pi would be an added advantage.

What you will learn from this book
- Set up the Raspberry Pi and the Cooking Hacks Arduino bridge shield
- Work with electronic components to build small circuits
- Develop applications that interact with your circuits
- Use thermistors, FSRs, and photoresistors to monitor and control your surroundings
- Communicate with your Raspberry Pi from an Arduino Uno via Ethernet
- Send yourself e-mail updates when a parcel arrives at your home
- Control your own mains power with a relay shield and even connect a motor to your Raspberry Pi
- Build a device that checks for damp/flooding using the Arduino bridge shield and Raspberry Pi


Andrew K. Dennis

Unleash the power of the most popular microboards to build convenient, useful, and fun home automation projects
In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 1 "An Introduction to the Raspberry Pi, Arduino, and Home Automation"
- A synopsis of the book’s content
- More information on Raspberry Pi Home Automation with Arduino Second Edition

About the Author

Andrew K. Dennis is the manager of professional services software development at Prometheus Research. This company is a leading provider of integrated data management for research, and is the home of HTSQL, an open source navigational query language for RDBMS.

Andrew has a diploma in computing and a BS in software engineering. He is currently studying for a second BS in creative computing.

He has over 10 years of experience in the software industry in the UK, Canada, and USA. His experience includes Python and JavaScript development, e-learning, CMS and LMS development, SCORM consultancy, web development in a variety of languages, open source application development, and a blog dedicated to maker culture and home automation.

His interests include web development, e-learning, 3D printing, Linux, the Raspberry Pi and Arduino, open source projects, parallel computing, home automation, amateur electronics, home networking, and software engineering.

Many of these topics were covered in his previous book, Raspberry Pi Super Cluster, Packt Publishing.
I would like to thank my wife, Megen, for supporting me throughout this project, and my parents for their support with my interest in technology while I was growing up.

I would also like to thank the team at Prometheus Research for making this a great and interesting place to work and helping to change the face of data management.

Finally, I would like to thank everyone who bought the first edition of this book, and the team at Packt Publishing for commissioning this second edition.
Raspberry Pi Home Automation with Arduino Second Edition

The world of home automation is an exciting field that has exploded over the past few years with many new technologies in both the commercial and open source worlds. This book provides a gateway for those interested in learning more about this topic and building their own projects.

With the introduction of the Raspberry Pi computer in 2012, a small and powerful tool is now available for the home automation enthusiast, programmer, and electronic hobbyist. It allows them to augment their home with sensors and software.

Combining Raspberry Pi with the power of the open-hardware Arduino platform, this book will take you through several projects in which you will build electronic sensors, and introduce you to software that will record their data for later use.

We hope you will enjoy the second edition of Raspberry Pi Home Automation with Arduino.

What This Book Covers

*Chapter 1, An Introduction to the Raspberry Pi, Arduino, and Home Automation,* introduces the technologies used in the book and provides a conceptual background to the world of home automation.

*Chapter 2, Getting Started – Setting Up Your Raspberry Pi and Arduino,* is a guide to your Raspberry Pi, Arduino, and the Cooking Hacks Raspberry Pi to Arduino bridge shield.

*Chapter 3, Central Air and Heating Thermostat,* teaches you how to build a thermostat using the Arduino platform to control your central air conditioning and heating.

*Chapter 4, Temperature Storage – Setting Up a Database to Store Your Results,* shows you how to build a database to store your projects’ data, and explore it via your web browser.

*Chapter 5, Parcel Delivery Detector,* demonstrates a system using Arduino and Raspberry Pi that will alert you whenever a parcel arrives at your door.

*Chapter 6, Curtain Automation – Open and Close the Curtains Based on the Ambient Light,* teaches you how to integrate motors with your projects to open and close blinds and curtains using the skills learned in previous chapters.
Chapter 7, Water/Damp Detection – Check for Damp/Flooding in Sheds and Basements, helps you build systems that can check for humidity and water to indicate dampness and flooding.

Chapter 8, Wrapping Up, finishes the topic with some ideas for future projects.

Appendix, References, lists a collection of links to the resources used in this book and other interesting information.
An Introduction to the Raspberry Pi, Arduino, and Home Automation

This chapter provides an introduction to the Raspberry Pi, Arduino, Arduino to Raspberry Pi connection bridge, and the subject of home automation.

We’ll look at the history of the Raspberry Pi and how it came to being, as well as the Arduino platform, an open source microcontroller that provides developers with a means to interact with their surroundings through a variety of sensors and motors.

Next, we will explore the Arduino to Raspberry Pi connection bridge, a method of attaching an Arduino shield to the Raspberry Pi. We will wrap up the chapter by covering home automation and how technologies such as the Raspberry Pi have put the ability to build complex sensor-based systems in the hands of the open source community.

First, let's start by looking at what we will be covering in the coming chapters of this book. We have a number of exciting projects ahead that will slowly introduce home automation via technologies such as Raspberry Pi and Arduino. These projects include:

- Writing software to control hardware
- Building a thermometer using a thermistor
- Turning the thermometer into a thermostat using relays
- Controlling electric motors using a motor shield
- Writing software to store sensor data generated by your projects
As you go through each chapter in this book, you will gain a basic knowledge of building circuits and hardware for home automation projects. You will then learn to write software to both control your hardware projects and record the data generated by them. Finally, we will consider future projects that you can build with your newly acquired skills.

Our next step is to learn a little about the background of the technologies that we are going to use. We will start with the Raspberry Pi.

**History and background of the Raspberry Pi**

From the first vacuum tube computers to the tape and punch card machines of the 1960s and the first microprocessor mainframes of the 1970s, computing had very much been the preserve of large businesses and the research departments of universities. However, by the late 1970s, with the release of Apple II and seeds planted earlier by technologies such as the TV Typewriter and Apple I, this was rapidly changing.

By the 1980s, the public could buy low-cost home computers, such as the ZX Spectrum and Commodore 64, which hit the market and subsequently gave birth to a whole generation of amateur programmers. By the 1990s, these programmers, brought up on tinkering with their home computers and writing BASIC, were heading to academia and the computer industry, and helping to forge the dot-com boom with game, web, and business technologies.

The genesis of the Raspberry Pi is linked to this in many ways. A group of computer scientists led by Eben Upton at the University of Cambridge's computer laboratory in 2006 struck upon the idea of producing a cheap, educational microcomputer geared towards amateur computer enthusiasts, budding students, and children. The aim was to help provide the skills for future computer science undergraduate applicants that many of the applicants in the 1990s possessed. This was largely because home computers of the 1980s required programming and were open to hacking.

However, it would be another two years before the project became viable, and until 2012 before the Raspberry Pi was being shipped to the public.
The 2000s saw a huge growth in mobile computing technologies, a large segment of which was being driven by the mobile phone industry. By 2005, ARM—a British designer of CPU core components and by-product of the 1980s' home computer company Acorn—had grown to a state where 98 percent of mobile phones were using their technology. This translated into around 1 billion CPU cores. ARM technology later ended up featuring on the Raspberry Pi, with the ARM1176JZF-S processor core being used as a part of the Broadcom BCM2835 System-on-a-Chip (SoC).

During the same period, Eben Upton designed several concepts for the Raspberry Pi, and by 2008, thanks to a by-product of the increasing penetration of mobile phone technology, the cost of building a miniature, portable microcomputer, with many of the multimedia functions that the public was accustomed to, was becoming viable. Thus, the Raspberry Pi Foundation was formed to develop and manufacture the Raspberry Pi computer.

By 2011, the first Alpha Models were being produced and tested, and the public finally got to see what the Raspberry Pi was capable of. Demos of Quake III Arena and full HD 1080px video showed that the tiny computer could pack a big punch for low cost.

Finally in 2012, the Raspberry Pi was ready for public consumption. Two versions of the Raspberry Pi were manufactured, namely Model A and Model B, with B being released first.

Over the subsequent years, both A and B were upgraded, with the Models A+ and B+ being release and this was complemented with the introduction of the Raspberry Pi 2 in 2015.

For the projects in this book, you will need to use at least a Model B version of the board, or the more powerful Model B+ or Raspberry Pi 2 Model B if available.

Next, let's learn about the Arduino platform.

**History and background of the Arduino**

One of the most popular open-hardware products to have hit the market is the Arduino platform. Developed in Italy by Massimo Banzi and David Cuartielles in 2005, Arduino is an open-hardware technology coupled with a programming language and an **Integrated Development Environment (IDE)** based on the open source wiring software.
The Arduino platform allows the user to create custom hardware and applications that control it via its namesake programming language. Cheap and easy to use, Arduino is an alternative to expensive programmable cards and closed commercial systems. Therefore, it helped pave the way for home enthusiasts to build their own home automation projects.

Currently, there are several board models in the market with a wide range of sizes and components; for example, the Lily Pad allows enthusiasts to attach an Arduino board to clothing for textile-based electronic projects. These boards support a wide range of shields—Arduino-compatible electronic boards that can be plugged into it and expand its functionality. One particular extension has been the introduction of Ethernet shields and wireless XBee devices that allow communication with home networks and the Web. What makes Arduino suitable for amateur enthusiasts is that little or no knowledge of how electronic components are soldered is required to use its prebuilt shields. As the user becomes more comfortable with the technology, they can progress to building their own projects using the numerous kits and sensors available on the market.

This easy adoption has helped contribute to a number of websites and books dedicated to home automation projects that use this technology.

Throughout the following chapters, you will explore the use of Arduino alongside the Raspberry Pi. For the projects in this book, we recommend using the Uno board pictured as follows:
Next, we shall look at the Raspberry Pi to Arduino connection bridge, a method of connecting your Arduino shields to your Raspberry Pi.

**Raspberry Pi to Arduino shield connection bridge**

For two of the projects in this book, we will be using the Raspberry Pi to Arduino shield connection bridge. This is produced by Cooking Hacks, an offshoot of the Libelium wireless communications company based in Spain.

By connecting this device to our Raspberry Pi, we get to reuse our Arduino shields and components, and get the power and storage of the Raspberry Pi running behind it. This opens up many options for complex home automation projects that may need to host databases or run memory-intensive software.

The Cooking Hacks shield is connected to the Raspberry Pi’s **General Purpose Input/Output (GPIO)** pins. With the inclusion of the arduPi software, you will be able to communicate between your electronic devices, the Raspberry Pi’s operating system, and web-based projects.
The Cooking Hacks shield should look like this:

![Image of Cooking Hacks shield]

Image courtesy: Cooking Hacks

A wide range of Arduino shields can be used with the connection bridge. At the time of writing this book, the Arduino shield connection bridge was available for €40.

Many of these shields can be found on the Arduino website at [http://arduino.cc/en/Main/Products](http://arduino.cc/en/Main/Products).

The shields specifications can be also be found at [http://www.cooking-hacks.com/documentation/tutorials/raspberry-pi-to-arduino-shields-connection-bridge](http://www.cooking-hacks.com/documentation/tutorials/raspberry-pi-to-arduino-shields-connection-bridge).

By learning to use this shield, you will discover how you can extend the functionality of your Raspberry Pi so that it can mimic the Arduino microcontroller and reuse components from other Arduino projects that you may have attempted in the past or will build in the future.
Soldering

Soldering is the process of attaching electronic components using a heated metal filler (the solder), in order to allow electrical current to flow between them.

At this point, it is worth mentioning that practicing some soldering before you start building the projects in this book is worth the effort, but not strictly necessary. If you are a novice, do not worry because there will be minimal soldering.

Also, if you have any old PC hardware sitting around, such as a graphics card that is no longer in use, you can practice desoldering and resoldering the components until you become comfortable with the process. This will also help you to get used to the soldering iron and desoldering tool.

Creating software for the Arduino

After you have set up the Arduino microcontroller and Arduino shield, and plugged it into the Raspberry Pi, you will probably be wondering how to interact with it. After all, it has sensors and LEDs, but these are nothing without applications to control them in a meaningful manner.

There are many software languages available for building projects with, but we are interested in the Arduino programming language, C++, Python, SQL, and HTSQL.

- The Arduino programming language is a subset of C++ and provides us with a tool to program the Arduino microcontroller, Arduino-compatible shields, and the components connected to them. One benefit of using this technology is that there is a wealth of programs and libraries online that can be used for future projects. You will be using this language in the Arduino IDE and on the Raspberry Pi to write the core applications that will be reading data from sensors attached to your projects.

- The next language that we will be using is Python. Python is a high-level programming language developed in the late 1980s by Guido Van Rossum named after the popular comedy show Monty Python's Flying Circus. This language allows you to build web and database applications that can be used to process the output of Arduino programs. We will be using Python to build a web application that can process data sent to it and then insert it via SQL into a SQLite3 database.

- We will also be using SQL to build the database that our Python script connects to. In conjunction with the SQLite database management system, we will construct a repository to store some of the results from our projects, for example temperature data.
• Besides these languages, we will also be using Hyper Text Structured Query Language (HTSQL) to provide a web interface to our database that is easy to access via the web browser and query via HTTP.

Now that we have looked at our tools to build home automation systems, the Raspberry Pi and Arduino, let's look at what home automation is.

What is home automation?

Having picked up this book, you may already have an idea of what home automation is, but just in case you don't, we'll give you a brief overview of the subject and the open source technology that is driving many projects out there today.

Home automation is more than just a remote control for your TV. Examples include programming your DVD to record your favorite shows, setting the AC unit to turn on when the temperature reaches 76 degrees Fahrenheit, and installing a fancy alarm system that contacts the police in case of a break-in.

Also known as domotics (a portmanteau between domestic and informatics), home automation can be summed up as a mechanism of removing as much human interaction as technically possible and desirable in various domestic processes, and replacing it with programmed electronic systems. It is essentially automation of home activities.

History of home automation

Concepts for home and building automation were around for decades before becoming a reality, and were featured in the writing of the 19th century science-fiction author H. G. Wells, comics, and cartoons such as The Jetsons. American industrialist George Westinghouse helped to pioneer the AC electrical system, which the X10 home automation standard would later run on, and in 1966, the company that bears his name, Westinghouse Electric, employed an engineer who developed what could arguably be called the first computerized home automation system—ECHO IV.

The Electronic Computing Home Operator (ECHO) was featured in the April 1968 edition of Popular Mechanics and had been expanded from a set of spare electronics, both in the physical sense and the literal sense, to include computing its founder Jim Sutherland’s family household finances and storing their shopping lists, amid an array of other tasks.
The ECHO never went commercial, and through the 1960s, hobbyists and a number of large companies such as Honeywell toyed with the idea of computerizing homes. However, it was the 1970s, much as with personal computing, that saw the birth of the modern era of home automation technology.

X10 – a standard is born
The introduction of the X10 technology standard can be arguably described as the beginning of modern home automation technology. Conceived in 1975 by Pico Electronics, who later partnered with Birmingham Sound Reproducers, X10 laid out the framework to allow remote-controlled access of domestic appliances. The X10 standard was designed to allow transmitters and receivers to work over existing electrical wiring systems by broadcasting messages such as "turn off" and "turn on" via radio frequency bursts.

In 1978, X10 products began to make their way into stores, geared towards electronics enthusiasts, and shortly after, in the 1980s, the CP-290 computer interface made its way to the market for the Mattel Aquarius computer.

The CP-290 unit allowed computers to communicate with X10-compatible home appliances. Over the years, support for Windows and Mac was included, and it gave those interested in home automation the ability to program their lighting systems, thermostats, and garage doors from their home computers.

As revolutionary as X10 has been, it unfortunately has a number of flaws. These include:

- Wiring and interference issues
- Commands getting lost in transmission
- Limited scope of products supporting X10
- Limited scope of commands available
- Slow speed of signal transmission
- Lack of encryption
- Lack of confirmation messages without expensive two-way devices
By the late 1990s, home automation still hadn't penetrated the home market on a truly wide scale. However, the technological advancements of the dot-com boom were providing a whole new set of tools, protocols, and standards that addressed many of the flaws of the X10 standard.

The dot-com boom and open source – a new set of technologies

With the explosion of technologies that followed the birth of the Web in the 1990s, home computing and networking technologies were now available to the public and could be easily and cheaply installed at home. These technologies would later provide an ideal candidate to push the boundaries of what could be achieved by home automation enthusiasts, and provide the industry with the tools to build smart home appliances and systems. It was only a small step from PC-to-PC communication to appliance-to-PC communication.

Home networks running on Ethernet, and later on Wi-Fi, provided a mechanism that could allow computers and electronic appliances to communicate with one another across a home without needing to use the existing electrical wiring. In the case of Wi-Fi, no extra cabling was required.

As protocols such as FTP and HTTP became the norm to access information across the Internet, hardware developers saw the opportunity to leverage these communication technologies in open source hardware devices. While X10 appliances had no way of knowing whether a signal had been successfully sent without the purchase of costly two-way devices, web technologies provided a whole framework to return error codes and messages.

At approximately the same time when the Arduino platform was being developed, the first tablet computers were beginning to be released. From 2005 until now, there has been an explosion in mobile, tablet, and smartphone devices. This growth has been commonly referenced to as the post-PC era.

These devices have provided mobile computing platforms that can run complex software and be small enough to fit in the user's pocket. As a result of this, applications that allow the user to control consumer electronics, such as the TV, have been developed for the iPhone and Android.

Due to their size, portability, and low cost in some cases, they have provided the perfect platform to interface with home appliances and devices, and provided an extension to a medium the user is familiar with.
Alongside the explosion in hardware, there was an equivalent explosion in software. One particular product of interest that we will look at is the open source Android operating system.

Android OS is a Linux-based operating system geared towards mobile devices. As a part of the Open Handset Alliance—a consortium of 84 companies operating in the mobile sphere—Google backed and eventually purchased the Android mobile operating system. The aim has been to create an open source operating system that can compete with companies such as Apple, and provide a robust system that can work across multiple manufacturers' devices.

As a result of this, commercial manufacturers of home appliances have begun to embed the technology and software into their products, and a generation of smart devices has started to appear in stores around the world.

**Commercial products**

If you are interested in a smart refrigerator that can tell you the weather and keep track of your groceries, or an oven that can be controlled via your smartphone, then you are in luck. Products such as the Samsung RF4289HARS refrigerator running Android and the LG smart washing machine are paving the way for smart homes by embracing open source and web-based technologies. It is also not just appliances that are getting the makeover. Firms such as the Nest—a company founded by ex-employees of Apple—are developing smart thermostats.

Barcodes and QR codes on products now allow the consumer to scan them with their smartphones and download information about the item directly from the web. This can be extended to allow scanning and inventory management of products at home, recording data such as consumption dates of products in the refrigerator, and dynamically generating shopping lists.

This combination of hardware, software, and information now provides the potential for the home to become a part of "the Internet of Things" as quoted by Kevin Ashton.

Thanks to open source and open standard technology being used in these devices, it is easy to combine home-brewed projects built with the Raspberry Pi and commercial products by companies such as LG to build a smart home with a network of devices that can communicate with one another to combine the execution of tasks.

As we mentioned, home-brewed systems such as the Raspberry Pi can form part of this network. Let's now look at the effects of the arrival of the Raspberry Pi on the world of home automation.
**Arrival of the Raspberry Pi**

With the arrival of the Raspberry Pi and the Raspberry Pi to Arduino shield, there is now a set of open source technologies that combine the power of a PC, the communication and multimedia technologies of the Web, the ability to interact with the environment of a microcontroller, and the portability of a mobile device.

Coupled with the existing Arduino microcontrollers, we have the perfect set of tools to allow us to build cheap devices and systems for our homes. These devices can then interface with their commercial counterparts and can also be tailored for our own needs, while providing great tools to learn about technology.

For those familiar with the Arduino platform, the Raspberry Pi combined with its shields provides an all-in-one medium to create devices without the need for a separate PC or Mac—giving us an alternative to solutions that currently exist.

The ability of the Arduino Uno to communicate with the Raspberry Pi via an Ethernet shield over a LAN gives us flexibility in the types of projects that we can build. You will see this in the coming chapters where an Arduino Uno writes information back to a Raspberry Pi that is hosting a database.

Also, thanks to the Raspberry Pi’s mission of providing an educational tool for those interested in programming, the addition of the Arduino shield will provide a mechanism for those who wish to move from writing software that manipulates the Raspberry Pi to software that manipulates their environment and provides a pathway to learn about electronics. This could have the positive effect of bolstering the ranks of Homebrew and Maker clubs with an eye towards home automation, and lead to an ever-increasing diversity of tools being produced for the public.

**Summary**

In this chapter, we provided an overview of the Raspberry Pi and Arduino platform. We also looked at some of the existing technologies used in home automation and their history. While Sutherland’s ECHO IV filled a room in his house, the Raspberry Pi occupies space not much larger than a credit card.

Home automation now seems to be taking the next step to becoming widely adopted, and the Raspberry Pi neatly fits into this world by providing those who want to customize control of their devices with an easy and a cheap tool to achieve it, and also by expanding what can be done with Arduino technology currently out in the marketplace.

With this in mind, we will get started on our first project—setting up the Raspberry Pi. Here, we will install the necessary software to get our Raspberry Pi up and running, and install the Arduino IDE so that we can program our Arduino Uno.
Where to buy this book


Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

Click here for ordering and shipping details.