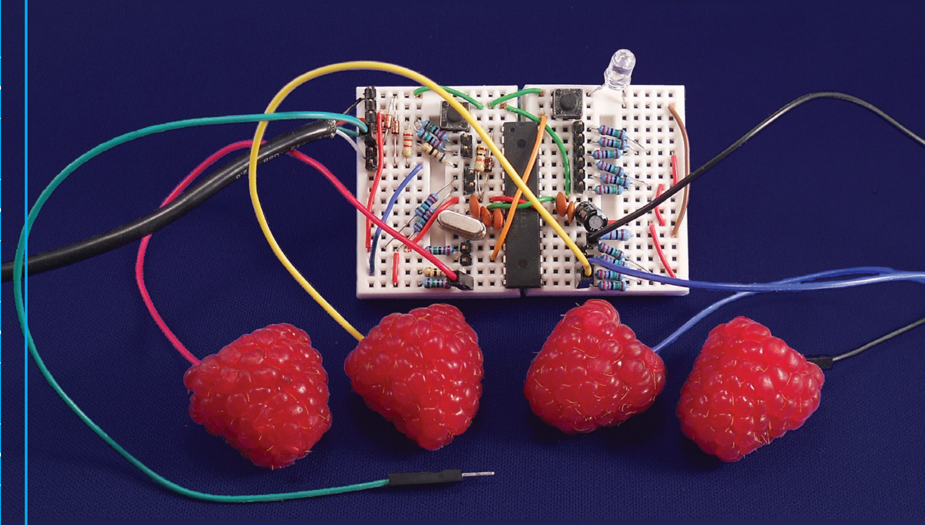
Make:

Raspberry Pi and AVR Projects



Augment Your Pi With Analog ICs and the Chip That Powers Arduino

Cefn Hoile, Clare Bowman, Sjoerd Dirk Meijer, Brian Corteil & Lauren Orsini

Make: Raspberry Pi and AVR Projects

Cefn Hoile, Clare Bowman,
Sjoerd Dirk Meijer,
Brian Corteil, Lauren Orsini

MAKER MEDIA™
SEBASTOPOL, CA

Make: Raspberry Pi and AVR Projects

by Cefn Hoile, Clare Bowman, Sjoerd Dirk Meijer, Brian Corteil, and Lauren Orsini

Printed in the United States of America.

Published by Maker Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

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 our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Editor: Patrick Di Justo
Production Editor: Melanie Yarbrough
Copyeditor: Charles Roumeliotis
Proofreader: Kim Cofer
Indexer: Angela Howard
Interior Designer: David Futato
Cover Designer: Brian Jepson
Illustrators: Cefn Hoile, Clare Bowman, Sjoerd Dirk Meijer, Brian Corteil, and Rebecca Demarest
Cover Photographer: Sjoerd Dirk Meijer

November 2014: First Edition

Revision History for the First Edition

2014-11-04: First Release

See *http://oreilly.com/catalog/errata.csp?isbn=9781457186240* for release details.

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.

978-1-457-18624-0

[LSI]

Contents

Preface

All of the projects in this book use the Raspberry Pi. Read along to discover how to receive Twitter alerts through a lamp, collect temperature data from a fish tank, control a game with Raspberries (the actual fruit!), or even learn how to make your own wearable one man band.

Book-wide bill of materials

Table P-1. *Book-wide bill of materials*

Item	Item
Raspberry Pi (Models B or B+)	Pi Cobbler
2x ADXL 345 triple axis accelerometer breakout board (labeled GY-291)	2x 100 ohm resistor
2x piezoelectric transducer	7x 170 point mini breadboards
2x Bluetooth UART transceiver (board name JY-MCU, running HC-06 firmware)	2x 1M ohm resistor
2x USB to UART serial programmer (CP2102)	3x 40-pin male header strip
2x ATmega328P-PU (flashed with Arduino Uno bootloader)	2x 5mm red ultrabright LED

Item	Item
6x 10k ohm resistors	4x 22pF ceramic capacitor
4x 100nF ceramic capacitor	Solid core wire
2x 4-way female ribbon cables	2x LiPo battery and charger or 6x AAA battery pack and batteries
Jumper wires male-male	Jumper wires male-female
DS18B20 digital temperature sensor	Push-to-make button/switch
3mm acrylic base plate	2x 2.5mm hex bolts, length 12mm
3mm acrylic top plate	2x 2.5mm nuts
4x PCB hex spacers 25mm	16x 3mm hex bolts, length 6mm
4x PCB hex spacers 10mm	4x sticky rubber feet
2x PCB round spacers 6mm	Double-sided sticky foam tape/pads
Wireless USB adapter	4.7k (1/4 Watt) resistor
Darlington array ULN2803A	Spotlight Kit LED RGB matrix
1x 400-point miniature breadboard	2mm heat shrink tube
Sparkfun Breadboard Power Supply 5V/3.3V part number PRT-00114	12V DC 2A power supply to 9V DC 2A power supply
1x 4-pole screw connector block for stripboard, w/ 0.1" pitch hole spacing	Wire 22 AWG
Stripboard prototyping PCB, w/ 0.1" pitch hole spacing	2.1 jack socket (breadboard compatible)
8GB SD card or 8GB SD card with NOOBS preloaded	2x button
1x ATmega328P-PU (DIP28) or ATmega8 (DIP28)	1x 10 uF capacitor
3x 100 nF/0.1 uF capacitor	2x 22 pF capacitor
16x 22 MΩ or 20 MΩ resistor (1/4W)	2x 10 kΩ resistor (1/4W)
1x 2.2 kΩ resistor (1/4W)	2x 68 Ω resistor (1/4W)
5x 330 Ω resistor (1/4W)	5x LED (5mm) (4 different colors)

Item	Item
1x 1N4148 diode	2x 3.6V Zener diode (max. 0.5W!)
1x 16 MHz crystal	

Table P-2. *Optional materials*

2x Copper stripboard 93 x 54mm	2x 28 pin DIP IC Socket Adapter
2x 40 pin male right angle header strip	ESD Anti Static Wrist Strap
1x stripboard (94 x 55 mm)	1x 28 pin DIP IC socket
Alligator clips	Small breadboard
1x 150 ohm resistor 1/4 watt through the hole	3x super bright green LEDs 20mA
1 x 315 ohm resistor 1/4 watt through the hole	3x super bright red LEDs 20mA
1 x 105 ohm resistor 1/4 watt through the hole	3x super bright blue LEDs 20mA

Table P-3. *Tools list*

Long-nosed pliers	Wire cutter
Wire stripper	Hot glue gun and glue sticks
Solder	Dedicated spot cutter or 4mm drill bit
Stripboard	Junior hacksaw
Soldering iron and holder	Permanent pen
Blue-tack (poster putty)	Ruler
G-clamp	Multimeter
Small vise	Saw
Hand drill	Hammer
Allen key to suit bolts	Stripboard cutter
Desoldering braid	Safety glasses
Solder sucker	Screwdrivers to suit
Craft knife	Masking tape

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic

Indicates new terms, URLs, email addresses, filenames, and file extensions.

`Constant width`

Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

`Constant width bold`

Shows commands or other text that should be typed literally by the user.

`Constant width italic`

Shows text that should be replaced with user-supplied values or by values determined by context.

This element signifies a tip, suggestion, or general note.

This element indicates a warning or caution.

Using Code Examples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from Make: books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Make: Raspberry Pi and AVR Projects* by Clare Bowman, Brian Corteil, Cefn Hoile, Sjoerd Dirk Meijer, and Lauren Orsini (Maker Media). Copyright 2014, 978-1-4571-8624-0."

If you feel your use of code examples falls outside fair use or the permission given here, feel free to contact us at *bookpermissions@makermedia.com*.

Safari® Books Online

Safari Books Online is an on-demand digital library that delivers expert content in both book and video form from the world's leading authors in technology and business.

Technology professionals, software developers, web designers, and business and creative professionals use Safari Books Online as their primary resource for research, problem solving, learning, and certification training.

Safari Books Online offers a range of product mixes and pricing programs for organizations, government agencies, and individuals. Subscribers have access to thousands of books, training videos, and prepublication manuscripts in one fully searchable database from publishers like Maker Media, O'Reilly Media, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, Course Technology, and dozens more. For more information about Safari Books Online, please visit us online.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

Make:
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States and Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

Make: unites, inspires, informs, and entertains a growing community of resourceful people who undertake amazing projects in their backyards, basements, and garages. Make: celebrates your right to tweak, hack, and bend any technology to your will. The Make: audience continues to be a growing culture and community that believes in bettering ourselves, our environment, our educational system—our entire world. This is much more than an audience, it's a worldwide movement that Make: is leading—we call it the Maker Movement.

For more information about Make:, visit us online:

Make: magazine: *http://makezine.com/magazine/*
Maker Faire: *http://makerfaire.com*
Makezine.com: *http://makezine.com*
Maker Shed: *http://makershed.com*

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at: *http://bit.ly/make-raspberry-pi-and-avr-projects*.

To comment or ask technical questions about this book, send email to: *bookquestions@oreilly.com*.

About the Authors

Clare Bowman enjoys hacking playful interactive installations and co-designing digitally fabricated consumer products. She has exhibited projects at Maker Faire UK, Victoria and Albert Museum, FutureEverything, and Curiosity Collective gallery shows. Some recent work includes "Sands Everything," an interactive hourglass installation interpreting Shakespeare's Seven Ages of Man soliloquy through gravity-controlled animated grains, and more.

Brian Corteil has never grown up, and still loves playing with computers, micro electronics, Lego, and video games. His first computers were a ZX80 then a TI-99, and finally an Acorn Electron. He is one of the founding members of Makespace, the place to make, fix, break stuff, and meet great people in Cambridge.

Cefn Hoile sculpts open source hardware and software, and helps others do the same. After 10 years of industrial R&D, Cefn founded *www.shrimping.it* to help schools and hobbyists source and adopt electronics prototyping materials in the classroom. One of the project's freely licensed builds, "The Shrimp," is an Arduino-compatible breadboard layout on which the ShrimpKey and Picussion projects in the book are based. Cefn is currently completing a PhD in Digital Innovation at Highwire, University of Lancaster, UK.

Sjoerd Dirk Meijer is the maker of ShrimpKey (DIY MakeyMakey) and a Scratch programming educator. He is also interested in (primary) education, giftedness, and making/Maker Ed. He can be found on Twitter @fromScratchEd.

Lauren Orsini is a technology journalist in Washington, DC. She writes about developer issues, tech education, and DIY hardware hacking for ReadWrite. Her new book, *Otaku Journalism: A Guide to Geek Reporting in the Digital Age*, is a new media journalism handbook to navigating Internet-age reporting.

Acknowledgments

Clare Bowman and Cefn Hoile

At OSHCamp (Open Source Hardware Camp) in Hebden Bridge, UK, we got to chatting with a man whose daughter had hyposensitive autism, a sensory problem where she is under-sensitive to stimuli. She found it difficult making friends as she did not know how to judge the personal space that others expected, and wanted to understand when she was getting too close.

We discussed a wearable device that would help her identify this personal space, loosely involving an ultrasonic module and vibration motor, housed in a necklace or belt that his daughter could design with a jewelry maker, so it was aesthetically pleasing and personal to her. Sadly we lost touch, and never found out if he managed to get the device built (if you are reading this book, we'd love to hear from you).

Through this encounter we started thinking about wearables, and at a body percussion workshop soon afterward, we realized how wireless wearable technology could turn your body into a drum machine, and Picussion was born. The UK Raspberry Jam network introduced us to the Pi, which is perfect for this emerging application, and the design began to take shape. Fast forward a few years, after a baby and a lot of experimentation, and we have included it in this book. Hope you enjoy building the project and your noisy journey with Picussion!

We want to thank the following people who have been immensely supportive while working on this project.

Firstly, Liz Edwards, a PhD student at Highwire, Lancaster University, who has been a keen contributor to Chapter 2, and her design skills have been invaluable. Thank you so much for your support and friendship. Paul Bowman, an inventor, cracking photographer, and wonderfully talented man in many other ways. Rachel Diss, an inspirational and supportive occupational therapist. The fabulous One Sam Band who performs around the North West of England. Our daughter Sky has put up with both parents hunched over laptops and sprawling circuitry while writing this chapter, so thank you Sky for being amazing, being our human guinea pig, and making us laugh when we were feeling tired and squinty-eyed. @ShrimpingIt thanks Fritzing.org for the use of vector graphics elements in the Picussion and ShrimpKey build diagrams. Thanks to Manu Brueggemann for camerawork and hand jiving. Also thanks to all our friends who have helped look after Sky and fed us cake while writing this chapter.

Brian Corteil

I first heard about the Raspberry Pi about three years ago, when a video about the prototype went viral. Back then I saw a small PC that would be cheap enough to leave in my projects, allowing them to connect to the Internet at below the cost of an Arduino Uno and Ethernet shield. I followed the project, waiting for the day that it would be released. When the day came, the release of the Raspberry Pi took down two of the UK's major electronic component suppliers' websites, but I was one of the lucky few who managed to order one. As of the writing of this book, the Raspberry Pi has sold over three million units!

I have a thing about pretty lights; RGB LEDs are candy to me. I love the way light can shape and bring out the detail in objects. So when I went to my local B&Q store, a child's night light caught my eye. The night light was in the shape of a giraffe, and my wife Sarah loves giraffes. She has a collection of at least 200, so as I turned the lamp over, I started formulating a project. I wanted a lamp that would create a "bat signal" for my wife to contact me when it flashed, since she has a habit of leaving her mobile phone in her purse with her coat thrown on top.

There are a number of people I would like to thank for their help and support.

First to the most important people in my life, Sarah, Billy, and Tom. Sarah, thank you for being the inspiration for this project and putting up with me as I have been writing this book when I should be doing stuff around the house. Billy and Tom, thank you both for letting me work on this project and your understanding, when I should have been playing with you both. It is my greatest joy watching you both grow up, I know you're going to be fine men that I will be proud to say are my sons.

I would like to thank the past and present members of Makespace for their support, help, and encouragement in the building and coding of this project. I wouldn't have been able to finish this project without them.

Sjoerd Dirk Meijer

Computers aren't just tools that support our daily lives, they're 21st century tinkering materials; you can make them do what you want (and not the other way around). But for that we have to learn the possibilities of computers and computer programming.

That's why I wanted to teach my daughter and son programming and, after searching the Internet, I found the graphical programming environment Scratch. It was exactly what I was looking for.

Most programming languages are English-based; so children in non-English countries have a disadvantage. They have to learn to read and translate the programming language, learn the syntax of the programming language, and have to learn to think as a programmer. With Scratch they only have to learn to think as as programmer, because the code is on blocks and the code is translated into many languages.

Making computer programs is fun. But it's even more fun if you can control your program with a homemade controller. After viewing Jay Silver's TED talk "Hack a banana, make a keyboard" I knew this was the "tool" I was looking for. After playing with it, I decided to use the MaKey MaKey in the programming workshops I host. But I needed more than one. Unfortunately, the MaKey MaKey isn't cheap and buying 10 of them would be too expensive for me. In the same period I found the Shrimp. After reading that the MaKey MaKey is Arduino-based, I was certain that it would be possible to make my own (cheaper) "MaKey MaKey" with a Shrimp. And it was!

I already owned a Raspberry Pi. And even though I love this credit card–sized computer, I couldn't find a functional way for me to use it. Until I found Simon Walter's ScratchGPIO. It's a special version of Scratch on the Raspberry Pi that can control the GPIO pins. With ScratchGPIO the Raspberry Pi can, like the Arduino, be used to control LEDs, motors, etc. Now children (and grown-ups) can program their own physical projects, without using "difficult" programming languages, like Processing, Python, and C.

Special thanks go to:

Jay Silver and Eric Rosenbaum for making the MaKey MaKey and licensing it as open source and open hardware. Without them I wouldn't have made the ShrimpKey.

Cefn Hoile and Stephan Baerwolf for their great help on the ShrimpKey source code.

And, most importantly, a big thank you to my wife and kids. They had to cope with a husband/father who used all of his spare time to write and therefore couldn't give them the attention they deserve. You three are amazing!

Lauren Orsini

I am not a programmer, or at least I wasn't until the Raspberry Pi came along. I wouldn't dare do anything that'd risk breaking my pricey MacBook. I always wanted to learn how to work with hardware, I just never had the right tools.

Enter the Raspberry Pi, which took away all of my self-imposed barriers and excuses. It was too inexpensive not to try out. And once I started messing with it, I couldn't stop. The best part, of course, is the enormous community of amateurs and experts alike, all chatting on the Web about their projects, concerns, and solutions for Raspberry Pi development.

By contributing here, I am glad to give back to the community that made me love Raspberry Pi. I hope you pick up this book and—whether you are a beginner or seasoned coder, an engineer, developer, or a writer like me—catch the Raspberry Pi bug.

Special thanks go to:

My husband, John, was a huge part of this project. While I am an amateur, John has some background in object-oriented programming and helped me to understand and customize the syntax behind the commands this project requires. And while I definitely got frustrated at times, John never gave up.

My friend, Patches, a professional programmer who was invaluable to me during my first attempt ever at tackling MySQL. Patches also turned me on to D3JS so I wouldn't have to try to build a database-reading graph from scratch.

My mom, who dedicated her first published book to me right after I was born. Sorry it took 27 years for me to return the favor, Mom!

And of course, my betta fish. I originally designed this project for Fintan, an older blue-and-orange fish, and it was originally named after him. Fintan didn't live to see this section published, but you can see photos of him peppered throughout it. He had some big shoes to fill so I brought home Levi, a juvenile betta with dark coloring, to personally test the final draft of this project.

And special thanks goes to Nelson Neves for his review of this book.

Introduction

1

Eben Upton invented the Raspberry Pi. As a teacher, he noticed that engineering students were getting worse, not better, at basic computer science concepts. He realized that students were not tinkering with the guts of their computers the way he had with his childhood Amiga. His solution was to prototype a $35 hackable computer, which we now know as the Raspberry Pi. With something so cheap and simple to program, engineering students could more easily study computing concepts.

Millions of people outside of his classroom scrambled to get their hands on a Pi. What he had designed for his students was now adopted by makers, tinkerers, and creators all over the world.

The Power of the Pi

The Raspberry Pi is a small computer. It has Ethernet networking, USB ports for peripherals, HDMI and composite video outputs, and a headphone jack for audio. It also has a small number of General Purpose Input/Output (GPIO) pins that can be controlled with software to send and receive binary information to control electronic circuits and microcontrollers.

The Pi boots from an SD card that contains its operating system and programs. The "New Out Of Box System" (NOOBS) preconfigured SD card gives new users a simple user interface for installing various operating systems, including the Raspbian Linux distribution, recommended by the Raspberry Pi foundation.

At the time of this writing, the Raspberry Pi Foundation released a new Pi model called the B+. This model offers more GPIO pins, additional USB ports, better audio, and lower power consumption. All four projects that are detailed in this book work with the B+.

Raspbian

Raspbian Linux makes the Raspberry Pi's desktop visible using an HDMI or RCA monitor, though most people prefer HDMI for a pixel-perfect display. While the Pi itself is cheap, the cost of peripherals like a monitor, keyboard, mouse, and power supply will cost many times more than the device itself.

One way to dodge such extra purchases is to network your Pi. All you need are a standard USB power cable and an Ethernet cable. Plug the USB into a power source, and plug the Ethernet into a router (or a laptop/desktop computer that is configured for network sharing), and you're good to go. Once the Pi has booted, you can see its desktop using a remote access program like VNC. With this type of setup, you can use your Pi without a keyboard, mouse, or monitor—"headless," as it is known in the Raspberry Pi community. However, because it sends the desktop as a series of images over VNC, it is not appropriate for high–frame-rate applications such as video playback.

Who Is This Book For?

This is not a beginner's book, but effort has been made to walk even novice readers through the projects. An electronics and soldering background would be nice to have prior to starting these projects, but is not necessary.

Projects in This Book

Five makers from all over the world have come together to share a few of their projects and inspire readers. This book consists of four main projects: Picussion, Raspberries from Scratch, Internet of Fish, and a Giraffe Mood Lamp.

Picussion, in Chapter 2, is a wireless, wearable device, made by Clare Bowman and Cefn Hoile, that can be configured by a Raspberry Pi to trigger sound samples by knocking/hitting it or orienting it in space.

The **Raspberries from Scratch** in Chapter 3, made by Sjoerd Dirk Meijer, turns everyday, conductive materials into a homemade conductive keyboard. This project walks you through building the conductive keyboard, and then walks you through making two computer programs with the Scratch language.

Chapter 4's **Internet of Fish**, by Lauren Orsini, is a smart, communicative thermometer. With this project, your Pi texts your cell phone with any overheating or cooling emergencies, and it also feeds a local web server that reads and visualizes the thermometer's data. This project could be used in other ways, such as monitoring a swimming pool, or maintaining a perfect temperature for an at-home brewery.

The **Giraffe Mood Lamp** in Chapter 5 has the brains of a Raspberry Pi and the exterior of a lamp. This project, created by Brian Corteil, is able to receive Twitter messages by harnessing the Twitter API, and flashes when it finds a new message.