

Evil Genius™ Series

Arduino + Android Projects for the Evil Genius
Bike, Scooter, and Chopper Projects for the Evil Genius
Bionics for the Evil Genius: 25 Build-It-Yourself Projects
Electronic Circuits for the Evil Genius, Second Edition: 64 Lessons with Projects
Electronic Gadgets for the Evil Genius, Second Edition
Electronic Gadgets for the Evil Genius: 28 Build-It-Yourself Projects
Electronic Sensors for the Evil Genius: 54 Electrifying Projects
15 Dangerously Mad Projects for the Evil Genius
50 Awesome Auto Projects for the Evil Genius
50 Green Projects for the Evil Genius
50 Model Rocket Projects for the Evil Genius
51 High-Tech Practical Jokes for the Evil Genius
46 Science Fair Projects for the Evil Genius
Fuel Cell Projects for the Evil Genius
Holography Projects for the Evil Genius
Mechatronics for the Evil Genius: 25 Build-It-Yourself Projects
Mind Performance Projects for the Evil Genius: 19 Brain-Bending Bio Hacks
MORE Electronic Gadgets for the Evil Genius: 40 NEW Build-It-Yourself Projects
101 Outer Space Projects for the Evil Genius
101 Spy Gadgets for the Evil Genius, Second Edition
123 PIC® Microcontroller Experiments for the Evil Genius
123 Robotics Experiments for the Evil Genius
125 Physics Projects for the Evil Genius
PC Mods for the Evil Genius: 25 Custom Builds to Turbocharge Your Computer
PICAXE Microcontroller Projects for the Evil Genius
Programming Video Games for the Evil Genius
Raspberry Pi Projects for the Evil Genius
Recycling Projects for the Evil Genius
Solar Energy Projects for the Evil Genius
Telephone Projects for the Evil Genius
30 Arduino Projects for the Evil Genius, Second Edition
tinyAVR Microcontroller Projects for the Evil Genius
22 Radio and Receiver Projects for the Evil Genius
25 Home Automation Projects for the Evil Genius

Raspberry Pi[®] Projects for the Evil Genius[™]

Donald Norris



New York Chicago San Francisco Lisbon London Madrid
Mexico City Milan New Delhi San Juan Seoul
Singapore Sydney Toronto

Copyright © 2014 by McGraw-Hill Education. All rights reserved. Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a data base or retrieval system, without the prior written permission of the publisher.

ISBN: 978-0-07-181956-5

MHID: 0-07-181956-8

The material in this eBook also appears in the print version of this title: ISBN: 978-0-07-182158-2, MHID: 0-07-182158-9.

E-book conversion by Codemantra

Version 2.0

All trademarks are trademarks of their respective owners. Rather than put a trademark symbol after every occurrence of a trademarked name, we use names in an editorial fashion only, and to the benefit of the trademark owner, with no intention of infringement of the trademark. Where such designations appear in this book, they have been printed with initial caps.

McGraw-Hill Education books are available at special quantity discounts to use as premiums and sales promotions or for use in corporate training programs. To contact a representative, please visit the Contact Us page at www.mhprofessional.com.

McGraw-Hill Education, the McGraw-Hill Education logo, TAB, Evil Genius, and related trade dress are trademarks or registered trademarks of McGraw-Hill Education and/or its affiliates in the United States and other countries and may not be used without written permission. All other trademarks are the property of their respective owners. McGraw-Hill Education is not associated with any product or vendor mentioned in this book.

Raspberry Pi is a trademark of the Raspberry Pi Foundation.

Information contained in this work has been obtained by McGraw-Hill Education from sources believed to be reliable. However, neither McGraw-Hill Education nor its authors guarantee the accuracy or completeness of any information published herein, and neither McGraw-Hill Education nor its authors shall be responsible for any errors, omissions, or damages arising out of use of this information. This work is published with the understanding that McGraw-Hill Education and its authors are supplying information but are not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought.

TERMS OF USE

This is a copyrighted work and McGraw-Hill Education and its licensors reserve all rights in and to the work. Use of this work is subject to these terms. Except as permitted under the Copyright Act of 1976 and the right to store and retrieve one copy of the work, you may not decompile, disassemble, reverse engineer, reproduce, modify, create derivative works based upon, transmit, distribute, disseminate, sell, publish or sublicense the work or any part of it without McGraw-Hill Education's prior consent. You may use the work for your own noncommercial and personal use; any other use of the work is strictly prohibited. Your right to use the work may be terminated if you fail to comply with these terms.

THE WORK IS PROVIDED "AS IS." MCGRAW-HILL EDUCATION AND ITS LICENSORS MAKE NO GUARANTEES OR WARRANTIES AS TO THE ACCURACY, ADEQUACY OR

COMPLETENESS OF OR RESULTS TO BE OBTAINED FROM USING THE WORK, INCLUDING ANY INFORMATION THAT CAN BE ACCESSED THROUGH THE WORK VIA HYPERLINK OR OTHERWISE, AND EXPRESSLY DISCLAIM ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. McGraw-Hill Education and its licensors do not warrant or guarantee that the functions contained in the work will meet your requirements or that its operation will be uninterrupted or error free. Neither McGraw-Hill Education nor its licensors shall be liable to you or anyone else for any inaccuracy, error or omission, regardless of cause, in the work or for any damages resulting therefrom. McGraw-Hill Education has no responsibility for the content of any information accessed through the work. Under no circumstances shall McGraw-Hill Education and/or its licensors be liable for any indirect, incidental, special, punitive, consequential or similar damages that result from the use of or inability to use the work, even if any of them has been advised of the possibility of such damages. This limitation of liability shall apply to any claim or cause whatsoever whether such claim or cause arises in contract, tort or otherwise.

*To Karen, my lovely soul mate and most ardent supporter.
Her faith in me never wavers and is my core strength.*

About the Author

Donald Norris has a degree in electrical engineering and an MBA specializing in production management. He is currently teaching undergrad and grad courses in the IT subject area at Southern New Hampshire University. He has also created and taught several robotics courses there. He has over 30 years of teaching experience as an adjunct professor at a variety of colleges and universities.

Mr. Norris retired from civilian government service with the U.S. Navy, where he specialized in acoustics related to nuclear submarines and associated advanced digital signal processing. Since then he has spent more than 17 years as a professional software developer using C, C#, C++, Python, and Java, as well as 5 years as a certified IT security consultant.

Mr. Norris started a consultancy, Norris Embedded Software Solutions (dba NESS LLC), that specializes in developing application solutions using microprocessors and microcontrollers. He likes to think of himself as a perpetual hobbyist and geek and is always trying out new approaches and out-of-the-box experiments. He is a licensed private pilot, photography buff, amateur radio operator, avid runner, and, last but very important, new grandfather—here's to you, Hudson.

Contents

Preface

Acknowledgments

1 Introduction to the Raspberry Pi

Hardware

Software

Summary

2 LED Blinker Project

Prototype Board

Soldering

Accessing the GPIO Pins

apt-get Command

LED Project

Summary

3 MP3 Player Project

Prototype Connector

Portable Stereo Speaker

Audio File Formats

Hardware Switch Inputs

Thinking in RasPi Time

MP3 Project Requirements

Summary

4 Camera Controller

Project Requirements

Camera Shutter Operation

Electronic Flash

Optoisolator Interfaces

Sound and Laser Modules

Interface Connections

Time-Lapse Function

Summary

5 GPS

Introduction

The Basics of How GPS Functions

The Ultimate GPS Receiver

6 Earthquake Detector

Introduction

Seismology and Earthquakes

Analog-to-Digital Conversion

Serial Peripheral Interface

Connecting and Testing the MCP3008 with the RasPi

Seismic Data Analysis

Operational System

Summary

7 Home Automation

Introduction

Z-Wave Mesh Network

RasPi and Z-Wave Interface

SSH Login

Open Z-Wave Software

Summary

8 Home Security Controller

Introduction

Webcam Selection

Laser Trip Assembly

Motion Software Package

Webcam Viewing

Laser Trip Program

Additional Sensors

Summary

9 NFC/RFID Reader

Introduction

How NFC Works

Installing libnfc

Hardware Installation

Initial Checkout

Project Program

Future Expansion

Summary

10 1-Wire Weather Station

Introduction

Weather Station Design

1-Wire Protocol

1-Wire Sniffer

Set Up the Weather Station Hardware

1-Wire File System

Viewing the Weather Data

Packet Sniffing

Future Expansion

Summary

11 Local and Remote Logic Controller

Introduction

1-Wire External Control Board

1-Wire File System (owfs) Installation and Configuration

Linux, Hardware, and FUSE

Test Sequences for the 8 Channel I/O Board

Python Test Program

Sniffer Monitoring

Android Remote Control

Testing the Web Server with a Browser

Summary

12 Robotic Car: Part 1

Introduction

Overview

Chassis and Drive Motors

Servos

13 Robotic Car: Part 2

Introduction

Robotic Car Block Diagram

I2C Software

Bluetooth Software

Robot Car Program

Operating the Robotic Car

Future Expansions

Summary

14 Radon Detector

Introduction

Radioactivity and Radon

K2645 Geiger-Muller Counter

Initial Test Configuration

Building a Portable Radiation Detector

Operating the Portable Geiger Counter

15 Serving Up Time with an RTC and NTP

Introduction

Real-Time Clock (RTC)

RTC Software

Introduction to the Network Time Protocol (NTP)

Building a RasPi NTP Server

Summary

Index

Preface

This Raspberry Pi project book is not only about building a series of interesting projects but also about providing an education regarding the underlying project technologies. I am positive that my over-30-years' experience as a college educator forced me to ensure that readers could not only build the projects but also understand why they function as designed.

Building a successful project is rewarding unto itself, but understanding why and how it functions is far more important. The reader should expect a manifold increase in experience with the Raspberry Pi if a commitment is made to expend the time and energy to complete most, if not all, of the projects. I learned a lot while completing them; sometimes things worked out fine, while other times they were not successful. But that's the joy of experimenting. As Professor Einstein once stated, "Anyone who has never made a mistake has never tried anything new."

The joy of learning about and building projects is the core concept within this book. I designed and built all of the projects, and along the way gained a lot of knowledge about the Linux OS and how it really shines as an embedded development platform.

I will not recap the projects here other than to state that the complexity increases from the book's beginning to its end. And this is how it should be, as experience and confidence in dealing with Linux and the Python language are progressively gained by proceeding through the projects.

Experienced Linux developers should feel free to jump into any of the projects; however, there are useful hints and techniques sprinkled throughout the book, which might be missed by taking a selective approach to reading it. I have also tried to point out the constraints and limitations of the Raspberry Pi as I encountered them when designing and building the projects. Just keep in mind, a \$35 computer simply cannot meet all expectations.

One disclaimer that I feel is warranted relates to the Python programs. These programs, while fully functional for the respective projects, are probably not in their best form. As I tell my beginning programming students, there are many ways to develop functional programs. Some are better than others—not necessarily right or wrong. With this perspective, I tried to keep the programs simple and to the point, and to avoid any unneeded complexity.

All of the book's projects can be expanded and modified as desired. I strongly recommend that readers do so, as that is one sure way to understand the concepts and bolster skills. The ability to experiment has been described as one of the key attributes that modern employers are looking for in twenty-first century employees.

Donald Norris

Acknowledgments

I thank Karen for putting up with all my experiments and enduring all the “discussions” about the book’s projects.

I also thank Roger Stewart for his continued support and guidance as the sponsoring editor. He is the true champion for this book.

Thanks to Amy Stonebraker for her support as editorial assistant.

Thanks also goes out to Nancy Dimitry for her fine work as the project manager.

Finally, I would like to thank all the folks at the Raspberry Pi Foundation for creating the board and getting it to the marketplace.

Introduction to the Raspberry Pi

THIS BOOK WILL INTRODUCE you to the Raspberry Pi and provide 12 projects that will lead you through some simple, fundamental operations up to some fairly complex ones. The Raspberry Pi, despite its small size, is a fully functional computer capable of running a full-fledged Linux operating system. It is also the most inexpensive computer with this level of functionality that is presently available to the public.

The Raspberry Pi is a small board measuring 56 by 85 mm, about the size of a standard credit card. Nonetheless, it contains some very impressive functionality, as you will discover later in this chapter. This new board is shown in [Fig. 1–1](#).

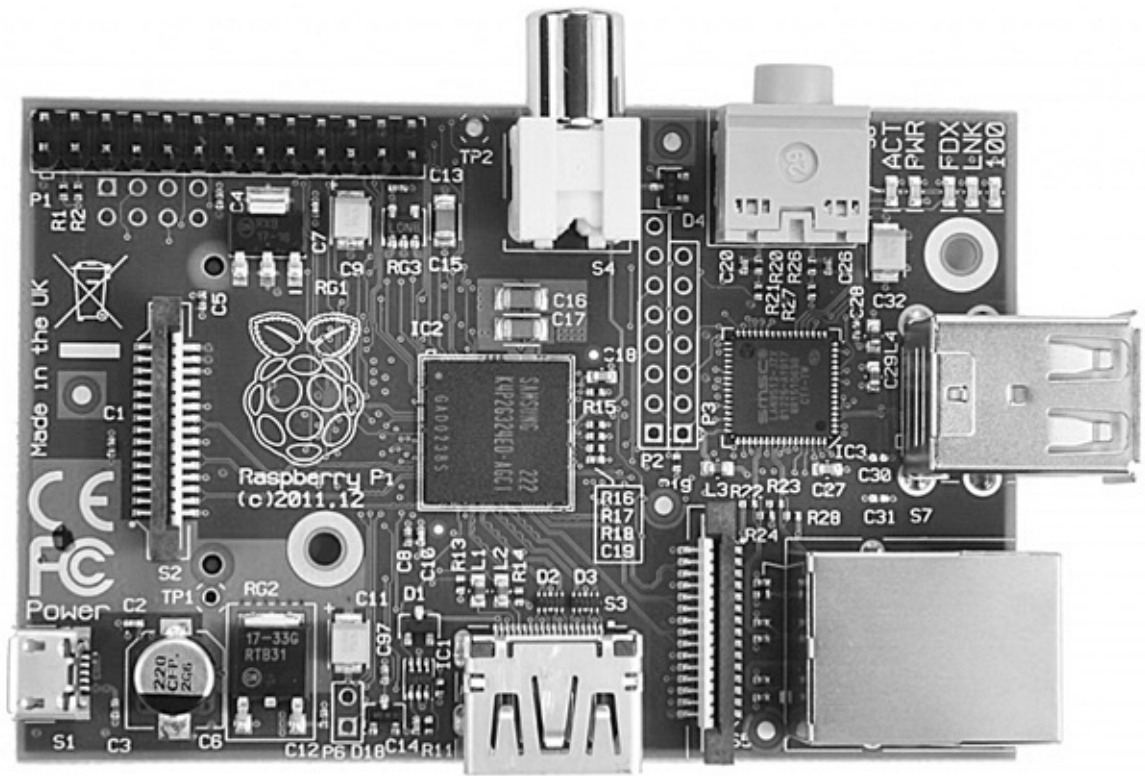


Figure 1–1 The Raspberry Pi, a small Linux computer.

A look at a bit of history regarding the Raspberry Pi (or RasPi as I will now refer to it) and its originally intended market may help you understand the constraints and limitations that subsequently ensued. The RasPi concept began around 2006 with Dr. Eben Upton and his colleagues at the University of Cambridge's Computer Laboratory in Cambridge, England. They were concerned about the decline in knowledge and skill levels of incoming computer science students as compared with those of earlier students. Dr. Upton decided to create an inexpensive computer, reasoning that it was likely that parents were not allowing their children to experiment with modern and relatively

expensive PCs. This idea ultimately led to the development of the very inexpensive RasPi. This computer would provide an excellent opportunity for children to learn and experiment with programming, while not being a concern to parents if something should go horribly wrong and the board be destroyed.

Dr. Upton teamed with several other individuals to form the Raspberry Pi Foundation, a registered United Kingdom charity that promotes computer literacy and enthusiasm, especially among young children using the RasPi as their initial platform. They seem to be achieving these highly laudable goals, since they have greatly exceeded the initial estimate of selling 10,000 RasPi's, and at the time of this writing, the total sales are approaching one million units. The foundation's website is www.raspberrypi.org, where you will find all sorts of information about the board, current news, forums, FAQs, and so on.

A key design decision that kept costs low was to incorporate a SoC type chip on the board. SoC is short for *System on a Chip*—a technology that physically places the memory, microprocessor, and graphics processor in a type of silicon “sandwich” that in turn minimizes the *printed circuit board* (PCB) space and the accompanying PCB interconnecting board traces. The foundation eventually partnered with Broadcom to use its designs for both the microprocessor and graphics processors in the SoC. The SoC and some other key components and connections that you should know about are identified in Fig. 1–2.

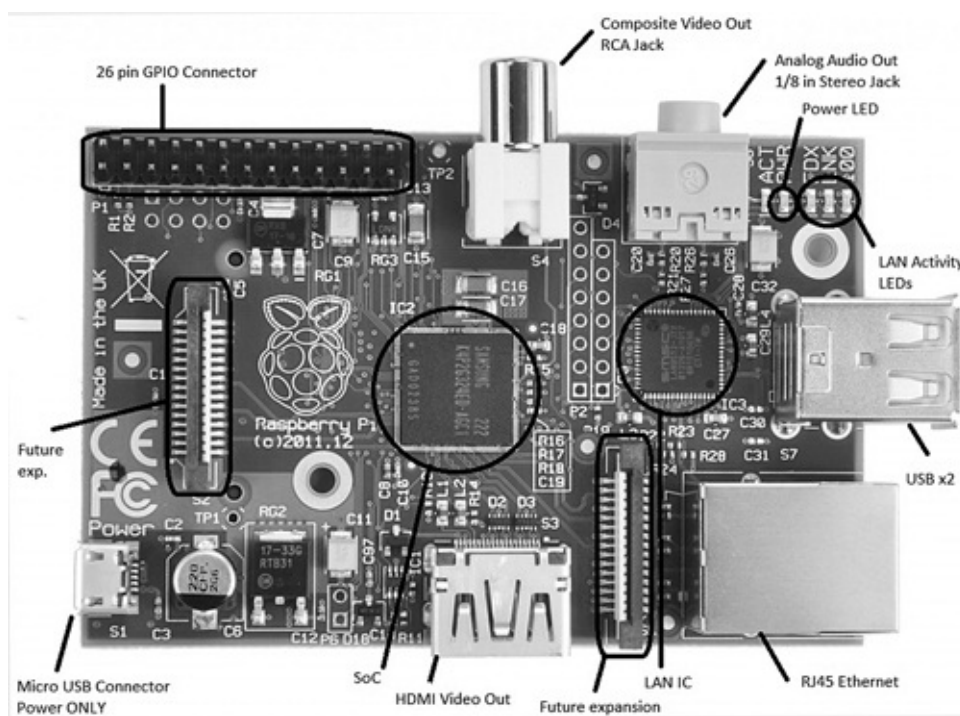


Figure 1–2 The SoC and other key components.

Although it is not critical to understand the Broadcom microprocessor in order to use the RasPi, it is still helpful to discuss it for a bit so that you will know why the RasPi is slower than your PC and why the low voltage of 3.3 V is used for interfacing the RasPi to the outside world. I will first cover the hardware aspects of the RasPi, followed by the software aspect.

Hardware

Broadcom 2835 Microprocessor/Graphics Processing Unit

The SoC uses the Broadcom BCM2835 as its microprocessor and *graphics processing unit* or GPU. The Broadcom company is what is known as a *fabless* supplier in that they provide the designs for their product in the form of *Intellectual Property* (IP) and other companies actually create the real silicon chips. Broadcom specializes in mobile-application-type processors including the type used in smartphones. The BCM2835 portion of the SoC itself is made up of an ARM1176JZF-S microprocessor running at 700 MHz and a Broadcom VideoCore® IV GPU.

The BCM2835 is designed for mobile applications, and hence, it needs to operate with minimal power so as to extend battery life. A fairly low microprocessor clock speed helps lower power consumption, and this is the reason the BCM2835 operates at 700 MHz, which is typically a quarter of the speed of a modern PC. Lower clock speed also means the processor can operate at a low voltage, thus decreasing the overall heat generated and extending chip life. The BCM2835 can be speeded up—also known as *overclocking*—to improve performance, but this is generally not recommended because the microprocessor can become operationally unstable and its life shortened. Be assured that the RasPi is sufficiently fast for all the projects in this book.

Broadcom has also graciously provided software drivers to allow the BCM2835 input and output pins to be connected to external peripherals. This software is in the form of a Python library that I will discuss later.

The Broadcom VideoCore IV GPU handles all the video and audio processing for the SoC. This GPU directly supports the OpenGL ES 2.0 standard that is essentially an *Application Program Interface* (API) capable of running on embedded hardware, which, in this case, is the Broadcom 2835. Loosely translated, this means the 2835 can easily display three-dimensional 3D graphics using all the requisite shaders and texture filters normally required for modern games and *high-definition* (HD) video. This chip implements in hardware a 1080p, 30 frames/sec, H.264 codec required for HD. That is an impressive performance.

For readers fascinated with performance statistics, it is interesting to note that the Broadcom Videocore IV GPU has the following processing capabilities:

1 gigapixel/sec (that's one billion pixels processed per second)

1.5 gigatexels/sec (that's one and a half billion texture elements per second)

24 gigaflops (that's 24 billion floating point operations per second)

All of this capability translates to the equivalent performance of a first generation Xbox®, not bad for a small embedded chip in a SoC sandwich!

I will not pursue this discussion any further other than to state that the BCM2835 is more than adequate to display all the graphics and output all the audio streams required for all the projects in this book.

Memory

There are two memory types used in the RasPi: *dynamic random access memory* (DRAM) and *Secure Digital* (SD) flash. The original version, model A, of the RasPi had 256 MB of RAM installed, while the latest, Model B, has 512 MB. The 512-MB chip is easily seen on the board as the top layer of the

SoC sandwich. The SoC chip shown in [Fig. 1-2](#) has top DRAM marked as supplied by Samsung with part number K4P4G324EB ACC1, which translates to a low-power 4-Gbit (512-MB) DRAM designed for mobile applications. This means that it too uses low voltage while maintaining reasonable clock speed. Having 512 MB of DRAM means the operating system will function very efficiently and programs should also run smoothly provided they are properly created.

The SD flash memory is used to store the operating system, all programs, and all other data that need persistence. In other words, nothing will be destroyed when the power is shut off. The RasPi uses SD flash memory in the same manner that a PC uses a hard drive to permanently store data and programs. You have a choice in selecting the memory size of a SD memory card that simply slides into a holder that is located on the underside of the RasPi, as shown in [Fig. 1-3](#).

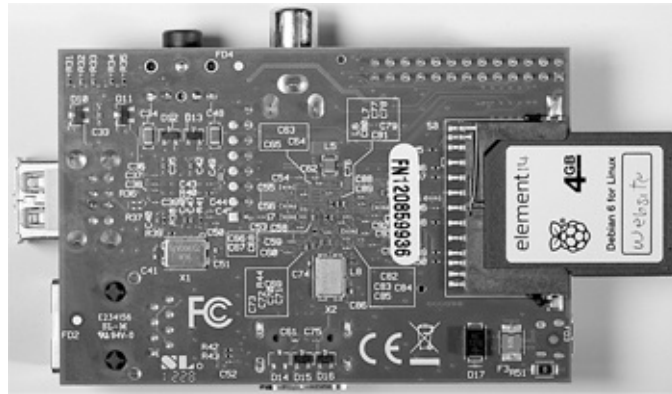


Figure 1-3 Back side of the Raspberry Pi.

If you purchased your RasPi as part of a starter kit, you will have received a 4-GB SD card with a Linux OS distribution already installed on the card. A picture of this preprogrammed SD card is shown in [Fig. 1-4](#).



Figure 1-4 Preprogrammed SD card.

I will explain later on in this chapter how to create your own memory card so you do not have to purchase a preprogrammed SD card. Note that most SD cards also have a class designation on the label. The one shown in [Fig. 1-3](#) does not, as it has had a customized label attached. The class designation relates to the minimum data-transfer speed the SD card can handle. Remember, the SD card is taking the place of a hard drive, so the faster, the better. Class 4 is very typical of the consumer-grade SD cards that are available in most office supply stores. Class designations and their associated minimum transfer speeds are shown in [Table 1-1](#).

Table 1-1 SD Card Class Designations

| Class | Minimum Performance |
|----------|---------------------|
| Class 2 | 2 MB/sec |
| Class 4 | 4 MB/sec |
| Class 6 | 6 MB/sec |
| Class 10 | 10 MB/sec |

What you should take away from this SD class discussion is that the higher the class number of the SD card used in the RasPi, the better it will perform. The only downside is that SD cards with high class numbers are more expensive than ones with lower numbers, sometimes more than twice the cost for the same storage capacity. My only suggestion is to purchase a class 4 or higher; anything less and you will be disappointed in your RasPi's slow response.

RasPi Connectors

The RasPi has nine connectors: power, *High-Definition Multimedia Interface* (HDMI), analog composite video, audio, ethernet, *Universal Serial Bus* (USB), two future expansion connectors, and the *General Purpose Input/Output* (GPIO) interface. Each connector has specific functions that I will discuss in the following sections, except for the expansion connectors, which are not yet used, although I will tell you what I know about them as of this writing. There is no particular order to this discussion although I have left the GPIO connector for last because it is by far the most complex and hence, requires the most explanation.

Power Connector

The power connector shown in [Fig. 1-5](#) is a micro USB socket that is wired to pass the 5-volt (V) *direct current* (DC) lines from a micro USB plug, also shown in the figure. (Since all voltage in this project is DC, I will use just the notation V for V DC.) No data connections are wired to this socket. You can use almost any available smartphone charger that has a micro USB connector or use the power supply that came with the RasPi kit, if that's what you purchased.

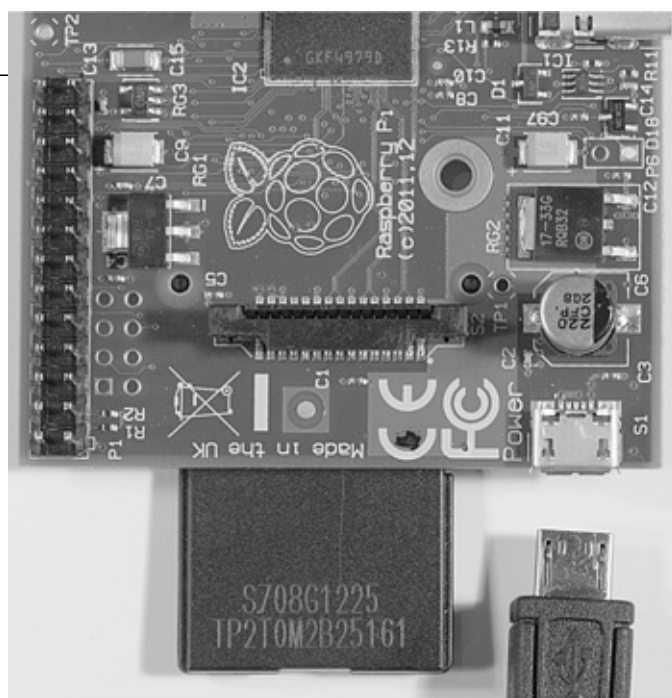


Figure 1-5 Micro USB power connector.

Figure 1-6 shows a RasPi kit power supply that is rated to supply 5 V at 1000 milliamperes (mA) or 1 ampere (A). The regulatory compliance document supplied with this RasPi states the following:



Figure 1-6 External power supply.

This product shall only be connected to an external power supply rated at 5 V, and a minimum current of 500–700 mA for model A and 700–1200 mA for model B.

I will have a bit more to say regarding current consumption when I discuss the USB connector.

HDMI Connector

The RasPi provides video and audio using a fully compliant HDMI, which is modern by most standards. The board socket and sample cable plug are shown in Fig. 1-7.

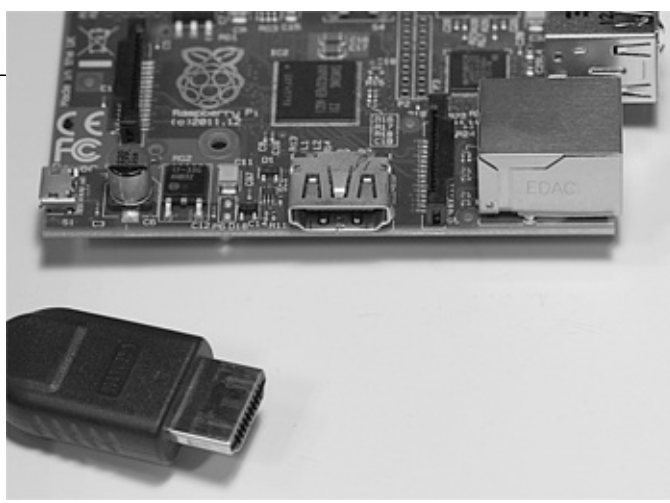


Figure 1-7 HDMI connector and cable.

I have previously discussed the Broadcom GPU chip that controls the HDMI output. To keep things simple, the book projects will use only the “standard” type of audio/video output and will not take advantage of the true potential of the RasPi’s multimedia capabilities. Trust a fellow Evil Genius that you will be working hard to complete the book projects without getting involved with HDMI development tasks.

One real problem that you will likely encounter is the lack of an HDMI input port for your computer monitor. This leaves you with three choices for observing the RasPi video:

1. Use the composite video output with a compatible analog monitor
2. Use an adapter to convert from HDMI to *Video Graphic Array (VGA)* or *Digital Video Interface (DVI)*
3. Take over the family’s flat panel digital TV

The first option is really not a very good choice, since the quality is diminished as compared to what is displayed by a high-quality computer monitor. The second option is the preferred method, as it yields the best results using your existing computer resources. Choosing the third and final option will likely result in family discord and upheaval for which I will take no responsibility!

The choice of an HDMI to VGA or HDMI to DVI adapter will, of course, depend upon what type of monitor input you have. Most monitors have a VGA input, and an adapter for that type of input is shown in [Fig. 1-8](#). The HDMI to DVI adapter is similar, and the cost for each is also similar.

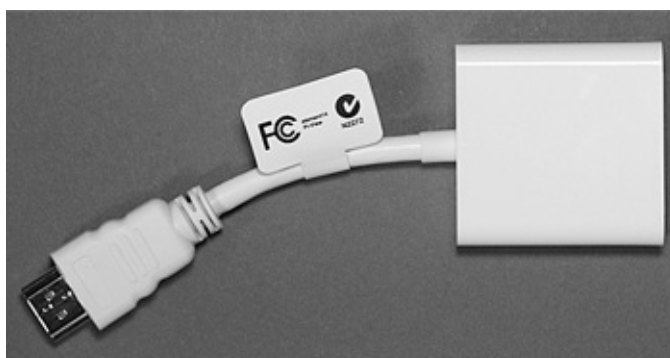


Figure 1-8 HDMI to VGA adapter.

The HDMI connection also contains a very interesting surprise. The RasPi can act as a very sophisticated remote control for HDMI-CEC compliant devices. The CEC suffix is short for *Consumer Electronics Control*, which is a one-wire, bidirectional serial bus link protocol used for the control of audio and video devices. HDMI-CEC has been implemented by many A/V manufacturers including Sony with its BraviaLink, LG with its Simplink, Sharp with its Aquos Link, Samsung with its Anynet+ and so on. The bad news is that there is currently no RasPi software support available for HDMI-CEC remote control functions. The good news is to simply wait for a short time because new software apps are constantly being created, free of charge. By the time you are reading this, the RasPi should be able to turn your digital flat-panel TV on and off as well as control your A/V receiver, DVD player, Blu-Ray player, etc. The RasPi will become the ultimate remote control. For more information, go to [http://elinux.org/CEC_\(Consumer_Electronics_Control\)_over_HDMI](http://elinux.org/CEC_(Consumer_Electronics_Control)_over_HDMI).

Analog Composite Video Connector

The RasPi also produces an analog video output from the RCA socket, as shown in [Fig. 1-9](#).

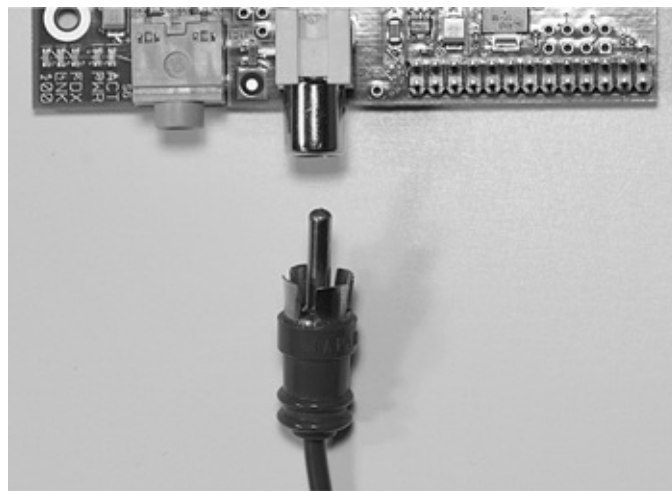


Figure 1-9 Analog video connector and cable.

This analog video functionality was deliberately included in the RasPi design to accommodate all those situations where only analog monitors or analog TVs are available, especially in developing countries. There is, however, an upside to the composite output. To monitor project parameters in real time, you can use small analog monitors. These monitors are fairly inexpensive and can often be battery powered, which is not a realistic option with larger computer monitors. I have included the use of a small, battery-powered analog monitor in one of the book projects. This monitor is shown in [Fig. 1-10](#).



Figure 1-10 Small analog video monitor.

Audio Connector

The RasPi is also capable of creating an analog audio output in full stereo. The output is from a standard 3.5-mm stereo jack as shown in [Fig. 1-11](#).

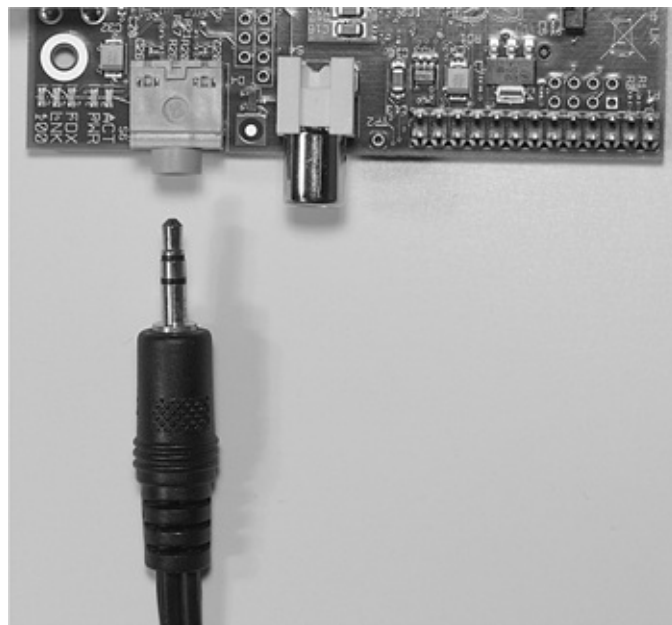


Figure 1-11 Analog audio connector and cable.

This audio would normally be the analog equivalent of the digital audio outputted from the HDMI connector. There is a book project that uses this analog output to play MP3 songs. You will need an audio amplifier to hear the music, as the RasPi does not generate a powerful enough signal to drive an unamplified speaker. However, a good quality set of headphones will work.

Ethernet and USB Connectors

Both the Ethernet and USB connectors are shown in [Fig. 1-12](#). I will discuss the Ethernet connector

first, followed by the USB connectors.

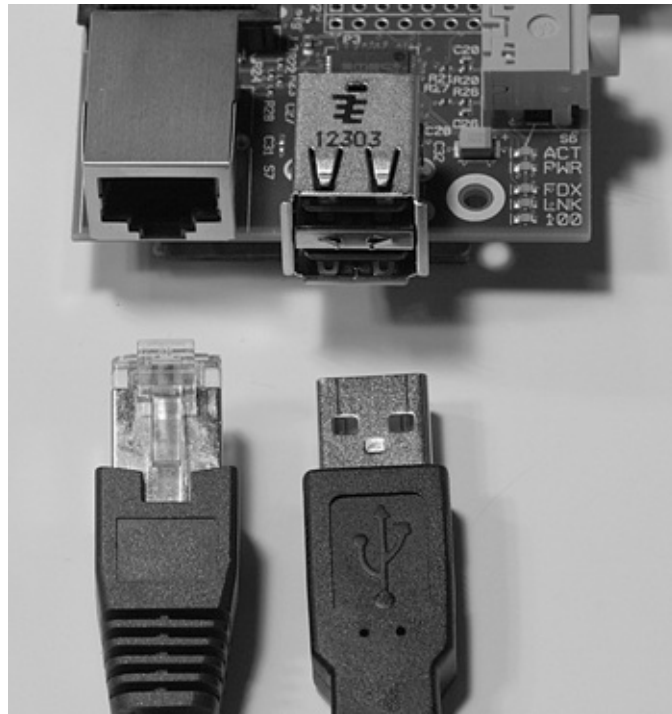


Figure 1-12 Ethernet and USB connectors with cables.

The Ethernet connector shown on the left in the figure is a standard RJ45 connector. You would simply plug your standard Ethernet patch cable into the socket and connect the other end to either your router or switch, if that is the way you have set up your home network. The RasPi will then automatically “negotiate” with your router to gain an *Internet Protocol* (IP) address in a process known as *Dynamic Host Configuration Protocol* (DHCP). There are five *light-emitting diodes* (LEDs) to the left side of this socket as you look at it head on. The LED furthest away from the socket is labeled “100.” If it is shining with a yellow light, this means that a 100-*megabits-per-second* (Mb/s) connection was made. The next two LEDs, just to the right of the 100 LED, are labeled “LNK” and “FDX”. These LEDs shine with green lights to indicate that the Ethernet is alive and operating. Checking these LEDs is a quick way to determine if your Ethernet connection is working or if something, somewhere, has gone down.

There is a stack of two USB connectors shown on the right-hand side of the figure. These are normal USB connectors in the sense that USB peripherals will be recognized when plugged into the sockets. The only issue with these is that the RasPi cannot supply the standard amount of current according to the USB specification, which is 500 mA per socket. Remember that I mentioned earlier in [Fig. 1-6](#) that the power supply in the RasPi kit provides up to 1000 mA. If peripherals plugged into these sockets took 500 mA each, there would be none left for the poor RasPi! Obviously, this situation should not be allowed to happen, and there is a good and relatively cheap solution. I use a powered USB hub, as shown in [Fig. 1-13](#), that can easily provide all the current that typical unpowered USB peripherals require.



Figure 1–13 Self-powered USB hub.

There is one USB cable that connects between the hub and the RasPi. That leaves one available USB socket on the RasPi for a low-power peripheral, such as a thumb drive. The number of USB ports provided by the hub varies with the manufacturer; however, four or five ports are fairly common. The power supply shown in the figure is rated for 2100 mA, which precisely matches the USB specification for four ports and a little left over for the hub internal electronics.

Future Expansion Connectors

Two connectors prominently populated on the RasPi are not currently used. Referred to as “future expansion” connectors, they reflect the dynamic nature of the RasPi project. [Fig. 1–14](#) is a close photograph of one of the connectors, labeled “S2.” This connector is a 15-way, flat-flex connector designated for use with the *Camera Serial Interface* (CSI-2). A prototype digital serial camera was just introduced at an international electronics show at the time of this writing. The other flat-flex connector labeled “S1” and located just behind the Ethernet RJ45 connector is designated as a *Display Serial Interface* (DSI) that will eventually drive a Sony *Low Voltage Differential Signaling* (LVDS) serial display. You should check the RasPi website for the latest news regarding the RasPi.

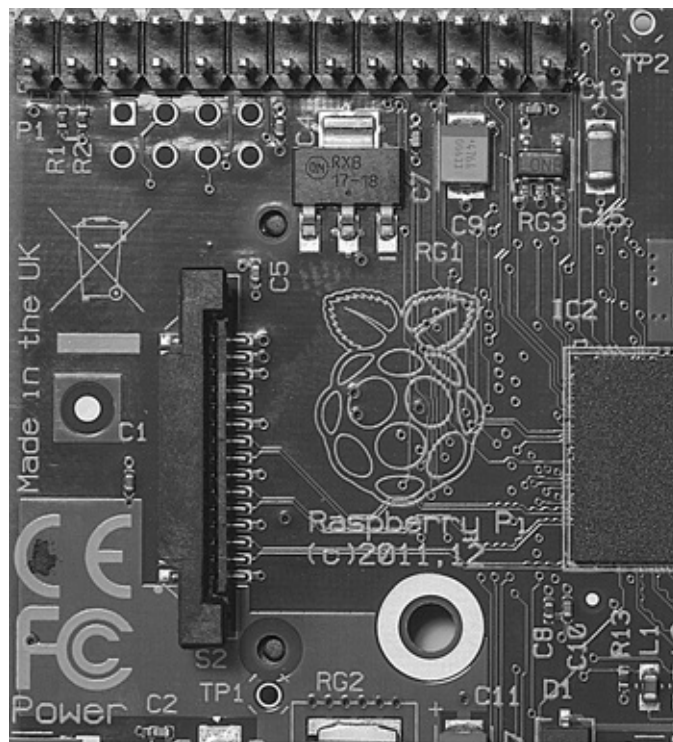


Figure 1–14 Expansion connectors.

GPIO Pin Interface Connector

- [read The Strategy of Campaigning: Lessons from Ronald Reagan and Boris Yeltsin.pdf, azw \(kindle\)](#)
- [click **Fairies and Fusiliers \(Barnes & Noble Digital Library\)**](#)
- [click **Hooked: Write Fiction That Grabs Readers at Page One & Never Lets Them Go**](#)
- [download online Artemis Invaded \(Artemis Awakening, Book 2\).pdf](#)
- [read **Greek and Roman Siege Machinery 399 BC-AD 363 \(New Vanguard, Volume 78\)**](#)

- <http://test.markblaustein.com/library/The-Java-Tutorial--A-Short-Course-on-the-Basics--5th-Edition-.pdf>
- <http://www.experienceolvera.co.uk/library/Hunter-s-Way--Hunter--Book-1-.pdf>
- <http://schroff.de/books/Working-Wonders.pdf>
- <http://www.experienceolvera.co.uk/library/Mutt--The-Rittenhouse-Saga--Book-1-.pdf>
- <http://betsy.wesleychapelcomputerrepair.com/library/Greek-and-Roman-Siege-Machinery-399-BC-AD-363--New-Vanguard--Volume-78-.pdf>