Extracted from:

# Arduino

## A Quick-Start Guide

This PDF file contains pages extracted from *Arduino*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit http://www.pragprog.com.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printer versions; the content is otherwise identical.

Copyright © 2010 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The Pragmatic Bookshelf

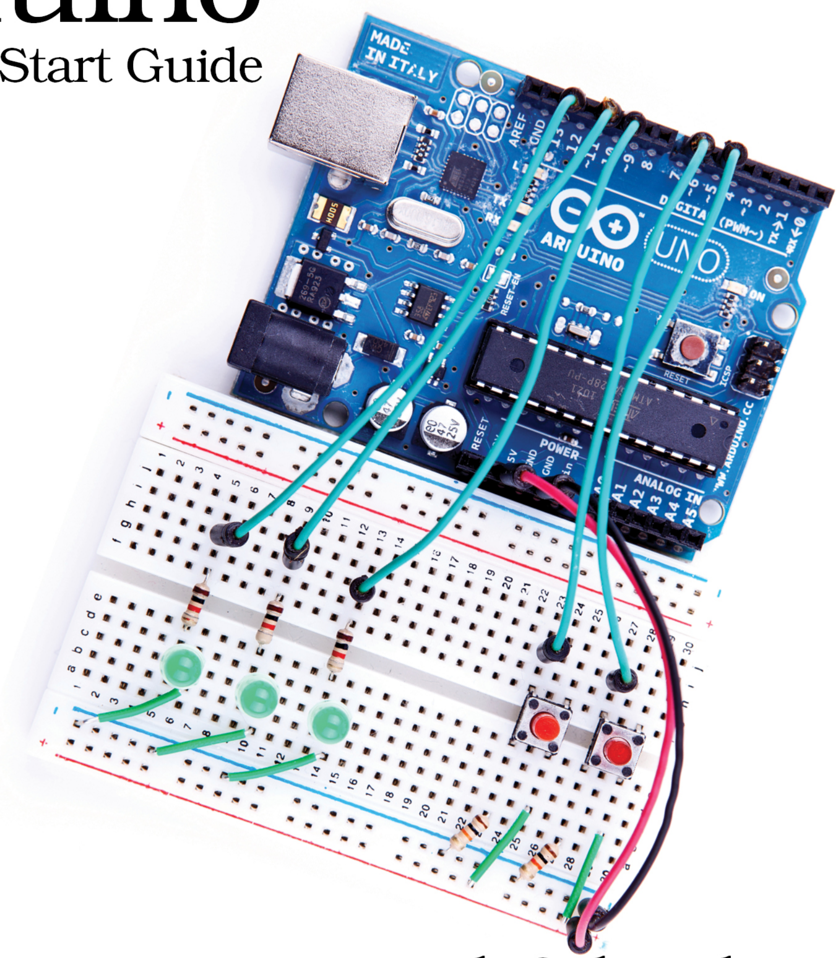
Dallas, Texas • Raleigh, North Carolina

The Pragmatic Programmers

For Arduino 1.0

# Arduino

## A Quick-Start Guide




Maik Schmidt

*Edited by Susannah Davidson Pfalzer*

One of the most entertaining electronic activities is simply tinkering: taking an existing product and turning it into something different or using it for an unintended purpose. Sometimes you have to open the product and void its warranty; other times you can safely make it part of your own project.

In this chapter, you'll learn how to hijack a Nintendo Nunchuk controller. It's a perfect candidate for tinkering: it comes with a three-axis accelerometer, an analog joystick, and two buttons, and it is very cheap (less than $20 at the time of this writing). Even better: because of its good design and its easy-to-access connectors, you can easily integrate it into your own projects.

We'll use an ordinary Nunchuk controller and transfer the data it emits to our computer using an Arduino. You'll learn how to wire it to the Arduino; how to write software that reads the controller's current state; and how to move, rotate, and scale a 3D cube on the screen using your Nunchuk. You don't even need a Nintendo Wii to do all of this—you only need a Nunchuk controller (see Figure 57, *A Nintendo Nunchuk controller*, on page 6).

## 7.1 What You Need

- An Arduino board such as the Uno, Duemilanove, or Diecimila
- A USB cable to connect the Arduino to your computer
- A Nintendo Nunchuk controller
- Four wires

## 7.2 Wiring a Wii Nunchuk

Wiring a Nunchuk to an Arduino really is a piece of cake. You don't have to open the Nunchuk or modify it in any way. You only have to put four wires into its connector and then connect the wires to the Arduino. (See Figure 58, *How to connect a Nunchuk to an Arduino*, on page 7.) It has six connectors, but only four of them are active: GND, 3.3 V, Data, and Clock. Put a wire into each connector, and then connect the wires to the Arduino. Connect the data wire to analog pin 4 and the clock wire to analog pin 5. The GND wire has to be connected to the Arduino's ground pin and the 3.3 V wire belongs to the Arduino's 3.3 V pin.

Here's the pinout of a Nunchuk plug:

• Click HERE to purchase this book now. discuss

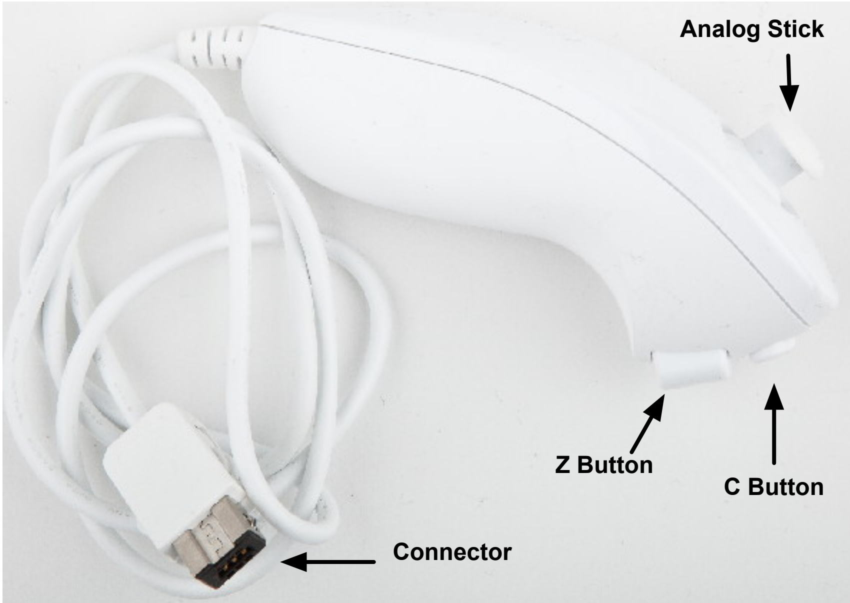
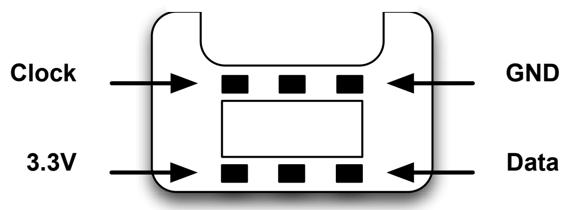



**Figure 57—A Nintendo Nunchuk controller**




That's really all you have to do to connect a Nunchuk controller to an Arduino. In the next section, you'll see that the two wires connected to analog pins 4 and 5 are all we need to interface with the controller.

## 7.3 Talking to a Nunchuk

No official documentation shows how a Nunchuk works internally or how you can use it in a non-Wii environment. But some smart hackers and makers on the Internet invested a lot of time to reverse-engineer what's happening inside the controller.[1]

1. http://www.windmeadow.com/node/42

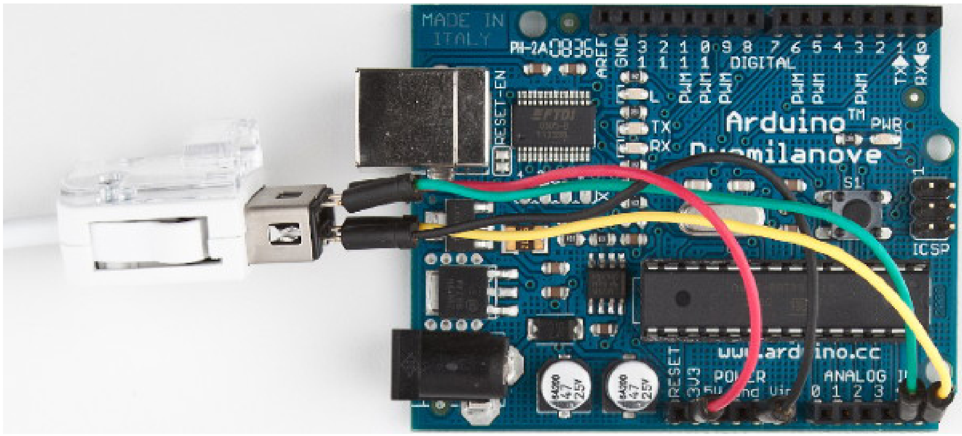



**Figure 58—How to connect a Nunchuk to an Arduino**

All in all, it's really simple, because the Nunchuk uses the Two-Wire Interface (TWI), also known as I$^2$C (Inter-Integrated Circuit) protocol.[2] It enables devices to communicate via a master/slave data bus using only two wires. You transmit data on one wire (DATA), while the other synchronizes the communication (CLOCK).

The Arduino IDE comes with a library named Wire that implements the I$^2$C protocol. It expects the data line to be connected to analog pin 4 and the clock line to analog pin 5. We'll use it shortly to communicate with the Nunchuk, but before that, we'll have a look at the commands the controller understands.[3]

To be honest, the Nunchuk understands only a single command: "Give me all your data." Whenever it receives this command, it returns six bytes that have the following meaning (see the data structure in Figure 59, *The Nunchuk always returns 6 bytes of data*, on page 8):

- Byte 1 contains the analog stick's x-axis value, and in byte 2 you'll find the stick's y-axis value. Both are 8-bit numbers and range from about 29 to 225.
- Acceleration values for the x-, y-, and z-axes are three 10-bit numbers. Bytes 3, 4, and 5 contain their eight most significant bits. You can find the missing two bits for each of them in byte 6.

2. http://en.wikipedia.org/wiki/I2c
3. At http://todbot.com/blog/2010/09/25/softi2cmaster-add-i2c-to-any-arduino-pins/, you can find a library that allows you to use any pair of pins for I$^2$C communication.

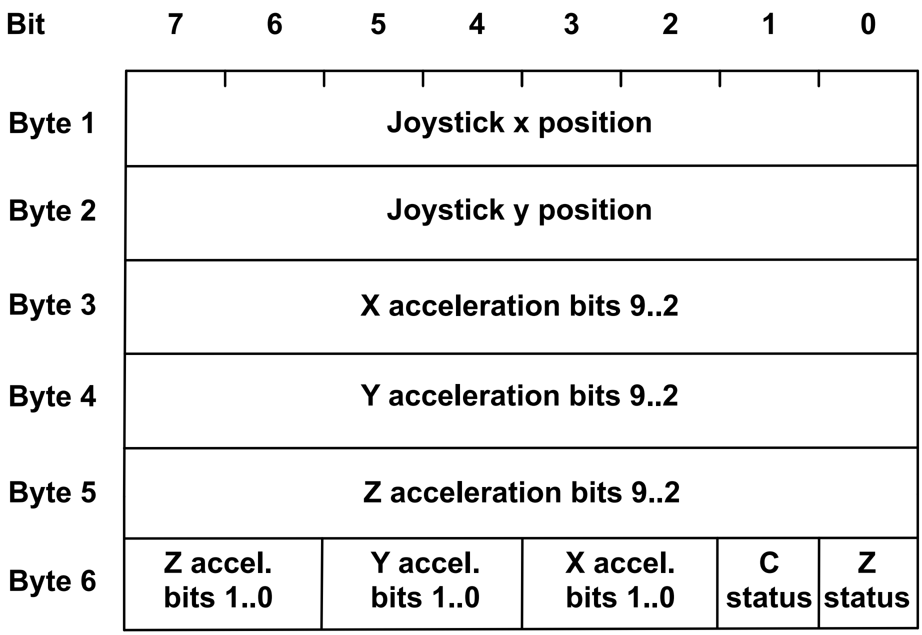



**Figure 59—The Nunchuk always returns 6 bytes of data.**

- Byte 6 has to be interpreted bit-wise. Bit 0 (the least significant bit) contains the status of the Z-button. It's 0 if the button was pressed; otherwise, it is 1. Bit 1 contains the C-button's status.

  The remaining six bits contain the missing least significant bits of the acceleration values. Bits 2 and 3 belong to the X axis, bits 4 and 5 belong to Y, and bits 6 and 7 belong to Z.

Now that we know how to interpret the data we get from the Nunchuk, we can start to build a `Nunchuk` class to control it.

## Improve People's Life with Tinkering

Because of its popularity, peripheral equipment for modern game consoles often is unbelievably cheap. Also, it's no longer limited to classic controllers; you can buy things like snowboard simulators or cameras. So, it comes as no surprise that creative people have built many interesting projects using hardware that was originally built for playing games.

An impressive and useful tinkering project is the Eyewriter.[a] It uses the PlayStation Eye (a camera for Sony's PlayStation 3) to track the movement of human eyes.

A team of hackers built it to enable their paralyzed friend to draw graffiti using his eyes. Because of a disease, this friend, an artist, is almost completely physically paralyzed and can only move his eyes. With the Eyewriter, he is able to create amazing artwork again.

It's not an Arduino project but definitely worth a look.

a. http://www.eyewriter.org/