

Wii Nunchuck as general purpose controller via Arduino board

by [XenonJohn](#) on May 2, 2010

Table of Contents

Wii Nunchuck as general purpose controller via Arduino board	1
Intro: Wii Nunchuck as general purpose controller via Arduino board	2
File Downloads	2
Step 1: View of Nunchuck buttons	3
Step 2: Arduino board inputs and outputs	3
Step 3: Plug on end of Nunchuck cable	4
Step 4: Connections between Nunchuck and Arduino	4
Step 5: Connecting Chuck to Arduino	5
Step 6: Connecting chuck to Arduino	5
Step 7: Cutting extender cable	5
Step 8: Pullup resistors	6
Step 9: More about the outputs	6
Step 10: This module in use	7
Related Instructables	7
Comments	7

Intro: Wii Nunchuck as general purpose controller via Arduino board

Background:

I made a self-balancing electric skateboard which is steered using a Wii Nunchuck by either using the thumb joystick on the top or by tilting it left or right, depending on which of the two buttons on the end are being pressed.

A simplified self balancing board is described in the instructable here:
www.instructables.com/id/Easy-build-self-balancing-skateboardrobotsegway/

However people are asking me how the Nunchuck steering control used on one of my earlier boards can be used. The wired Nunchuck can be seen in use as a steering control in the second half of this video:
www.youtube.com/watch

It can also be seen in use controlling a set of LED's attached to the Arduino outputs here:
www.youtube.com/watch

I have another instructable on interfacing a wireless Nunchuck to an Arduino but that is quite tricky and only works with a few specific wireless nunchucks. If starting out I strongly suggest going for the wired standard Nunchuck to begin with as described here.

Purpose:

The purpose of this instructable is to describe how to interface a completely standard Wii Nunchuck (i.e. with a cable) with an Arduino board. The Nunchuck is low cost but powerful with a fully proportional thumb joystick, 2 buttons on the end and a 3 axis accelerometer (position sensor). It is ideal for controlling things with.

The Arduino has been set up to output variable voltages between 0 and 5V which can be used as, for example, steering control inputs for another project such as a robot. You can use the (proper propotional) thumb joystick or the embedded accelerometers (tilt the chuck left right forward or back).

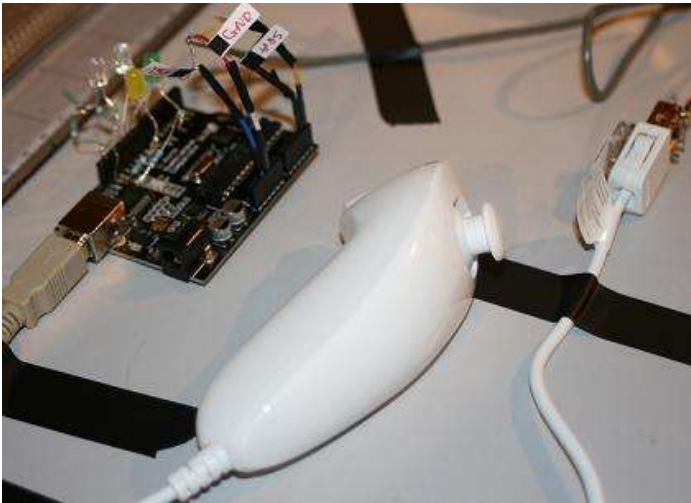
By modifying the code you can easily make it do other things.

I am using an Arduino with a 328 processor (i.e. a recent one) and version 17 of the Arduino software. The Arduino sketch is attached below.

This is all derived from code developed by Chad Phillips and others, see: <http://windmeadow.com/node/42>
For more info on wiring up the chuck to the Arduino, download Bionic Arduino Class 4 tutorial which is excellent from TodBot blog:
<http://todbot.com/blog/2007/11/24/bionic-arduino-class-notes-3-4/>

There is plenty more on this on the net, there is even someone with a Nunchuck controlled electric canoe here:
www.youtube.com/watch

I am just presenting this all together in one place as a way to get started with something that should work OK.



File Downloads



WorkingWiredNunchController2ndMay2010.pde (8 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'WorkingWiredNunchController2ndMay2010.pde']

Step 1: View of Nunchuck buttons

Here are the "c" and "z" buttons on front of the Nunchuck.



Step 2: Arduino board inputs and outputs

Outputs:

The outputs from the Arduino are arranged as follows;

Digital Pin 9: outputs +5V when either of the end buttons are pressed. This can be used as a go/stop control signal. If controlling a vehicle / robot or similar, then letting go of the Nunchuck will create a 0V at this output and so can be used to kill power to motors and so on.

Digital Pin 10: PWM (pulse width modulated) output on pin 10 gives 2.5V most of time. NOTE: This is not a smooth 2.5V but a series of pulses at 500Hz where half the time there is 0V and half the time there is +5V. If you smooth them out using a resistor and a capacitor (called an RC filter) you can use the resulting (now smoothed) voltage as a control input signal for some other device (another arduino or whatever).

Output is 2.5V when joystick in neutral position.

It climbs to +5V or drops to 0V when joystick moved *forwards* or *backwards*.

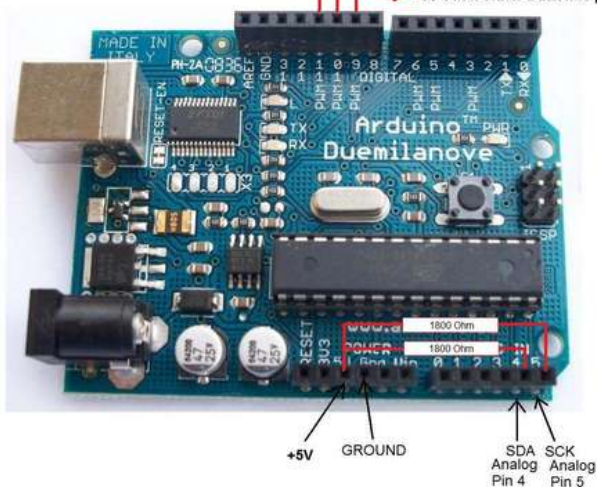
Digital Pin 11: PWM (pulse width modulated) output on pin 11 gives 2.5V most of time (in similar way to Pin 10 above) when joystick in neutral position. If z button is pressed, *then* joystick left/right will make it go down to 0V or up to 5V.

BUT: If both z *and* c buttons pressed (or just c button alone) then this Pin 11 output becomes controlled by Left/Right TILT of the nunchuck (via embedded accelerometer) and the L/R joystick component is disabled.

So: The output voltage (smoothed with an R/C filter) can be used as a steering input to some other device. If pressing z button, they you steer with joystick, if c button (or both) is being pressed, then you steer by tilting Nunchuck left or right.

2.5V out (PWM). If z button pressed, 0V if Joy left, +5V if Joy right. If c pressed, 0V if tilted left, +5V if tilted right.

2.5V out (PWM), 0V if Joy back, +5V if Joy forwards
+5V out if either button is pressed



Step 3: Plug on end of Nunchuck cable

This is the plug on the end of the Nunchuck cable.

On next page I will show a diagram of the connections you see when you look into the end of it.

There are 6 small copper contacts. We will be using the outer 4, the 2 middle ones are not used.



Step 4: Connections between Nunchuck and Arduino

Power to Nunchuck from Arduino board:

Use the 5V output on Arduino to Nunchuck (works better than using 3.3V line)

Connections between Nunchuck plug and Arduino:

5V output on Arduino to Nunchuck (works better than using 3.3V line)

GND (connect to ground on Arduino)

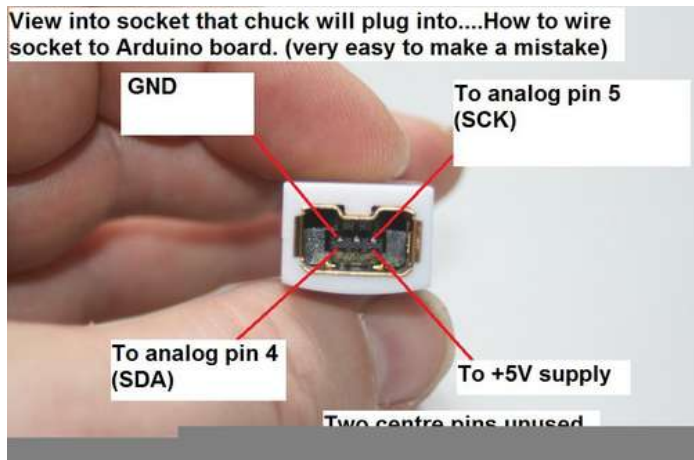
SDA connects to analog pin 4

SCK connects to analog pin 5

Note: the middle upper and middle lower contacts of nunchuck plug are not used

To connect the Nunchuck to th Arduino I buy a Nunchuck cable extender which has a female socket on one end of it. I cut the socket off with about 6 inches of wire hanging from it, bare the wire end and solder the wires to the arduino board. This way on your project, you just plug your Nunchuck into this female socket which you mount somewhere convenient.

I will explain how to actually do this on next page



Step 5: Connecting Chuck to Arduino

How to physically connect Nunchuck to Arduino:

You can get or make adapters that fit into plug on end of the Nunchuck cable. My worry is that these would work loose.

I prefer to buy a Wii Nunchuck "cable extender" – cheap on ebay.

This is a cable that has a socket on one end that the plug on the end of the Nunchuck fits into.



Step 6: Connecting chuck to Arduino

You are going to cut this cable in half and keep the socket this end as shown by the arrow.

The cut wires will be soldered to the Arduino board.



Step 7: Cutting extender cable

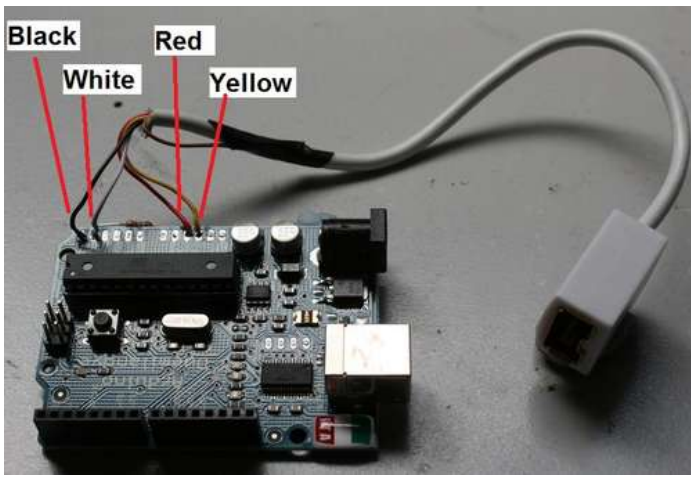
Cut the cable extender so that you have this socket plus a short length of cable on it. Expose the wires at other end to the socket and, using a voltmeter set to measure resistance, work out which colour wire is going to which of the 6 metal contacts inside the socket.

You are only going to use 4 of the 6 wires.

Work out which wire goes to which contact inside the female socket (referring to earlier diagram of the socket you have just made for the nunchuck to plug into) - check and check again - easy to get everything back to front), label the wire ends.

- +5V (power to chuck from +5V of arduino)
- GND (to ground pin on arduino)
- SDA (connects to analog Pin 4 on arduino)
- SCK (connects to analog Pin 5 on arduino)

Photo below show colours of wires in the cable extender I bought and where they go on the arduino board. These are correct only if you buy exactly the same cable extender as seen in previous photo. If you buy a different one the wire colours may be different so double check against my earlier socket pin wiring diagram.



Step 8: Pullup resistors

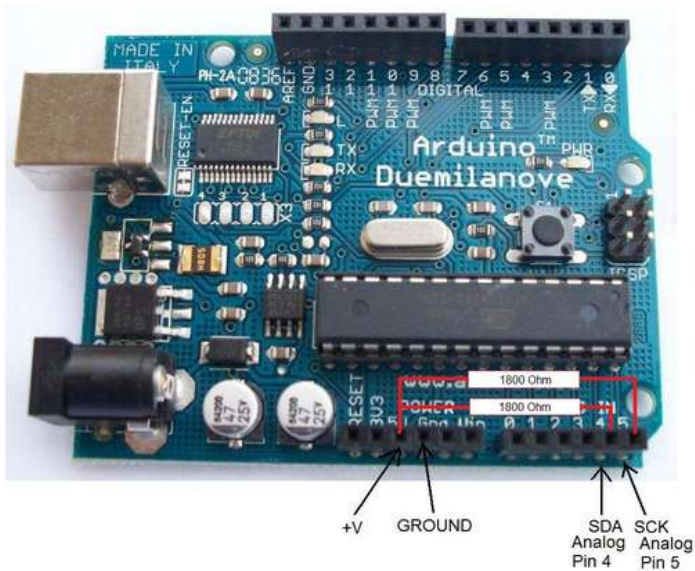
This is not essential, but some do it for reliability:

Connect a resistor (about 1800 Ohms worked for me) between analog Pin 4 and +5V on the Arduino.

Also, connect a resistor (about 1800 Ohms worked for me) between analog Pin 5 and +5V on the Arduino.

(This is more important for the wireless Nunchucks which we are not concerned with here and which require completely different software).

NOTE: This may change soon. Working on software that will automatically read either a wireless chuck or a wired Nintendo chuck, depending which one you plug into the socket.



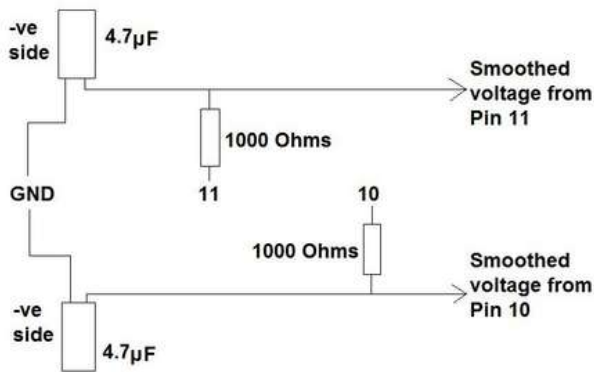
Step 9: More about the outputs

Output Pin 9 is either 0 or 5V, nothing in between. It can be used to cut power to motors for example if neither of the buttons are being pressed on the Nunchuck.

Output Pins 10 and 11 are producing a variable voltage. However this is as a series of pulses of 0 or 5V. To use these as a variable voltage control signal input to another device (as a steering command for example where 2.5V would be "straight ahead") we need to smooth the square wave pulses into a proper stable voltage.

For this we use an RC filter mentioned earlier.

A diagram of how this is set up for each output pin is shown here with values for the resistor and capacitor that I found to work OK.



Step 10: This module in use

I used the Nunchuck/Arduino combination as a steering input for a self-balancing skateboard. The variable voltage steering commands were fed into the inputs of an older non-Arduino microcontroller which was doing the "balancing" and controlling power to the two motors. It worked well and it can be seen in action in the second half of this video:

www.youtube.com/watch

Related Instructables



Using WiiChuck Adapter with Arduino by josestude



Wireless Wii Nunchuck controlled Arduino by XenonJohn



Angle measurement using gyro, accelerometer and Arduino by otaviousp



Control Your Robot Using a Wii Nunchuck (and an Arduino) by oomlout



Arduino Labyrinth by tripletray



Mario Bros. costumes with sound effects by RobHopeless

Comments

3 comments [Add Comment](#)



geekflytk says:

Funny english... I am gonna use this method to build my Segway Clone utilizing his other tutorial.

Sep 20, 2010. 10:36 AM [REPLY](#)



scouris says:

I love the 'english' on the packaging. While not in the same flair as hacking your own connection from the extension cable, you can actually buy wii adaptors that connect to the nunchuck, for very little. Here is a link to an Australian supplier - <http://www.littlebirdelectronics.com/products/WiiChuck-Adapter.html> but I'm sure that they are available elsewhere.

Aug 29, 2010. 11:06 PM [REPLY](#)



Simpson_jr says:

Hi,
A very interesting project, especially since I need a low cost solution ;-)

May 28, 2010. 2:57 AM [REPLY](#)

Unfortunately I still play all my games on a ZX81 (well, sort of), hit my head with a real nunchuck once and I have no experience with a Wii nunchuck.

In my project I'll need to measure parallelism and temperature changes in a solder-wave oven. When a PCB is tilted too much components on one side of the board won't get soldered properly.

Professional equipment is too costly for my cheapskate boss, but since I rather develop equipment as place leds all day I offered to look into a self built device.

So, I'll use the electronics of the nunchuck as a digital spirit level. One thing I'd like to know, what resolution in angle measurement (x and/or Y) does an intact nunchuck have ?

As far as I understand now I might be able to use the analog output of the accelerometer chip. By using an opamp-circuit i could get a high resolution at a small angle (PCBs never enter the oven at a big angle). But like my boss I'm a cheapskate as well and when reading the I2C-port would already give me enough resolution I'd be very happy.

I might even get promoted to placing transistors all day, I'm really looking forward to that, so any help will be greatly appreciated ;-)

Best wishes,
Bart

