

The 5\$ Karduinoss pad

by [buZztiaan](#) on December 16, 2009

Table of Contents

License: Public Domain Dedication (pd)	2
Intro: The 5\$ Karduinoss pad	2
step 1: Find the signals	2
step 2: Soldering to the touch-pad	4
step 3: Interfacing to the arduino.	4
File Downloads	5
step 4: Package it up	5
step 5: Use it!	5
Advertisements	6

Intro: The 5\$ KarduinoSS pad

So, looking at these Kaoss pads and alike hardware, I found that there is hardly any point in this device being so expensive, when you just want to use it as MIDI controller.

Going through my parts bin, I found a Synaptics touchpad from an old laptop and figured this should just work as a replacement.

Ok, when I say 5\$, i mean really cheap. Clearly a loose arduino is already 20-25 USD (but you can replace it with just a bare ATMEGA168 chip for ~2 USD) , and this touchpad would probably cost you some money aswell, when you are not able to salvage it from an old laptop.

WARNING: the Kaoss pads and similar hardware all have built-in audio outputs, this project does not ...

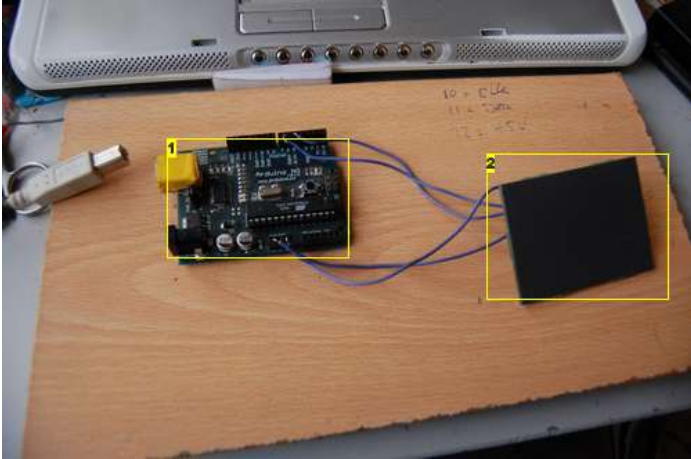


Image Notes

1. Arduino NG
2. Synaptics Touchpad

step 1: Find the signals

After some brief google (and going through the mostly pointless documentation) I found this website that helped me a great deal :

<http://sparktronics.blogspot.com/2008/05/synaptics-t1004-based-touchpad-to-ps2.html>

This website had the picture you see here. From this I figured out which of the testpoints on the board (wow, many!) were connected to these 3 pins. (see second picture).

I'm not quite sure if the T1001 controller is similar in pin-out to this controller, but the signals shouldn't be too hard to find if you know what to look for.

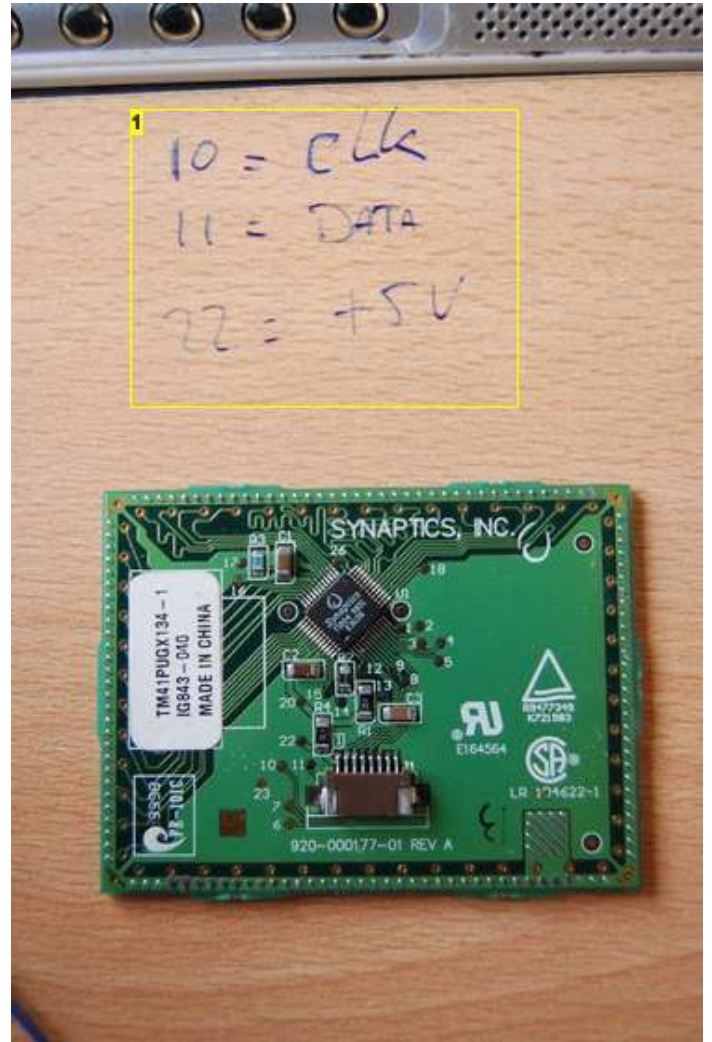


Image Notes

1. Be sure to write down what you find ;)



Image Notes

1. Yeah not the greatest :D

step 2: Soldering to the touch-pad

Well, this is simple enough, bring out the points you have found :D

The main trick is to use pretinned wires and first get a small island of solder on the testpoints you have found. Be careful to not put too much heat on the touch-pad, the copper can very easily get loose from the pad by overheating.

For ground I soldered to the big square connection (see also the pin-out image), but there are many more places where you can get ground signal.

After this step, you probably want to secure your soldering with some hot glue or similar product for strain relief.

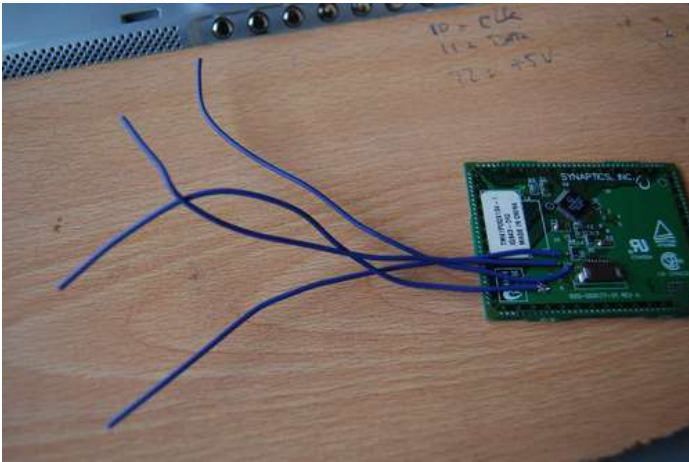


Image Notes

1. Yeah not the greatest :D

step 3: Interfacing to the arduino.

As you now have a touch-pad with wires attached to it, lets look into getting values from it.

The available ps2 libraries for arduino naturally work as a mouse and allow for mouse type input, if you're into such a thing ;)

Below you will find a karduinoss.pde sketch for the arduino based on the ps2 library available on <http://www.arduino.cc/playground/ComponentLib/Ps2mouse> which will initialize the touch-pad as an absolute xy controller with about ~4000 steps side to side.

The code does some auto-calibration based on the values it gets, and maps the x, y and z values to MIDI controller changes through the use of ttymidi available on <http://www.varal.org/ttymidi/>.

The karduinoss.pde sketch assumes an LED is connected to pin 3, the touch-pad clock on pin 13 and the touch-pad data on pin 12.

```
File Edit Sketch Tools Help
ps2_mouse
#include <ps2.h>

/*
 * an arduino sketch to interface with a ps/2 mouse.
 * Also uses serial protocol to talk back to the host
 * and report what it finds.
 */

/*
 * Pin 5 is the mouse data pin, pin 6 is the Clock pin
 * Feel free to use whatever pins are convenient.
 */
PS2 mouse(6, 5);

/*
 * initialize the mouse. Reset it, and place it into remote
 * mode, so we can get the encoder data on demand.
 */
void mouse_init()
{
  mouse.write(0xFF); // reset
  mouse.read(); // ack byte
  mouse.read(); // blank
  mouse.read(); // blank
  mouse.write(0xF0); // remote mode
  mouse.read(); // ack
  delayMicroseconds(100);
}

Done uploading.
Binary sketch size: 3492 bytes (of a 14336 byte maximum)
1
```

Image Notes

1. The ps2 library's ps2_mouse.pde example

File Downloads



karduinooss.pde (2 KB)

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

step 4: Package it up

Well, let your imagination run wild :D

I save plastic containers from random stuff to use for projects like this.

The packaging I made clearly is not the most solid possible, but I'm sure it will hold for a couple of performances before I need to think about a new case.

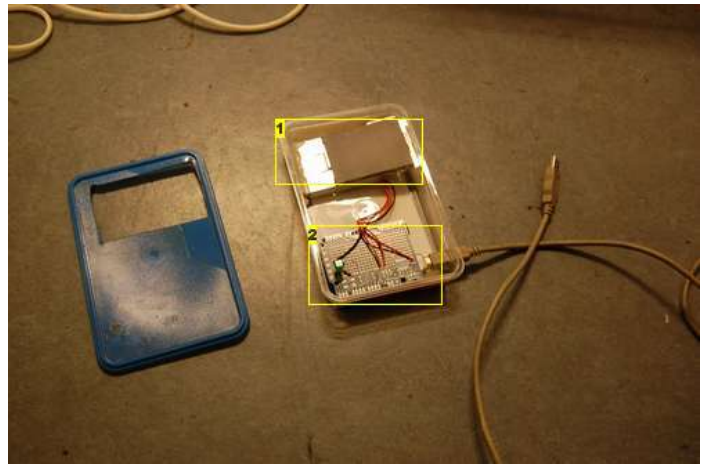


Image Notes

1. The touchpad, encased in a piece of metal from the laptop I salvaged it from
2. Arduino and a protoshield (for a more solid connection to the arduino)

step 5: Use it!

Now is the time to actually use your newly made karduinooss pad for some nice midi action :)

Below you see a screenshot of 'alsamodular synth' (which you can find on <http://alsamodular.sourceforge.net/>)

But of course you are free to use it in any program that supports midi input :D

I hope everybody enjoyed this instructable and will find inspiration to make his/her own midi controller now!

