Touchpad Pinout Testing

This is from step 5 of my KVM Instructable

This write-up will focus on PS/2 touchpads but USB, Serial, ADB (Apple), or TTL are also used in laptops. Newer touchpads are powered with 3.3 volts but 5 volts is more common for an old thick laptop like the one you'll use for this project. PS/2 is an open drain (aka open collector) clock and data bus that relies on pull up resistors to set the logic high level. My code bit-bangs two of the Teensy I/O pins to create a PS/2 clock and data bus per the Synaptics touchpad interface protocol given in documents 511-000275-01 and 510-000080. Any movement or button pushes are translated by the Teensy from PS/2 to USB mouse commands. You will need to locate the PS/2 connections on the touchpad so they can be wired to the Teensy. Google your touchpad part number to see if any pinout information exists. If not, search for your laptop's schematic on line to identify the pinout, pull-up resistors and supply voltage at the motherboard touchpad connector. Use an ohm meter to trace the power, ground, clock, and data signals thru the touchpad cable to the connector on the touchpad.

The following websites may be helpful in determining the pinout of your touchpad.

http://diycommunity.4fan.cz/pinouts/

https://www.instructables.com/id/Laptoprecycling-Touchpad/

https://www.instructables.com/id/Turn-a-Trackpad-From-a-Broken-Laptop-into-a-PS2-M/

https://www.instructables.com/id/The-5-Karduinoss-pad/

https://www.youtube.com/watch?v=XdznW0ZuzGo

https://www.youtube.com/watch?v=XSUJaHIhJMQ

https://www.youtube.com/watch?v=IFb7LYIHWN0

http://domoticx.com/pinout/

http://pinoutguide.com/Inputs/alps_touchpad_pinout.shtml

Synaptics touchpads often have round test points that are perfect for attaching wires. The typical test point numbers for a Synaptics PS/2 touchpad are:

- Clock = T10. If T10 is not present, use T12
- Data = T11. If T11 is not present, use T13
- Ground = T23 or solder to the Ground Plane
- Power = T22. This test point provides an RC power filter. You can skip the power filter and tie directly to the power pins on the Synaptics chip at T20.

If your Synaptics touchpad has numbered test points but T10/T11 or T12/T13 are not found, the touchpad may use USB or some other protocol.

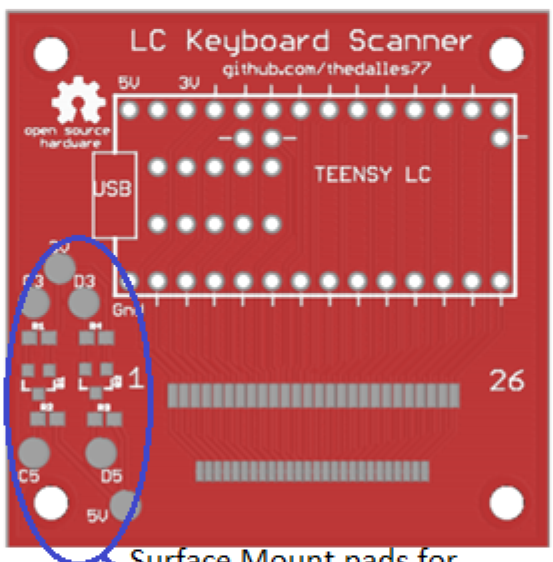
The Teensy 3.2 has 5 volt tolerant I/O and will work with a 5 volt or 3.3 volt touchpad. No level translators are needed. If you add pull up resistors, they should go to the touchpad supply voltage.

The Teensy LC is not 5 volt tolerant and can be ruined if the I/O pins go to 5 volts.

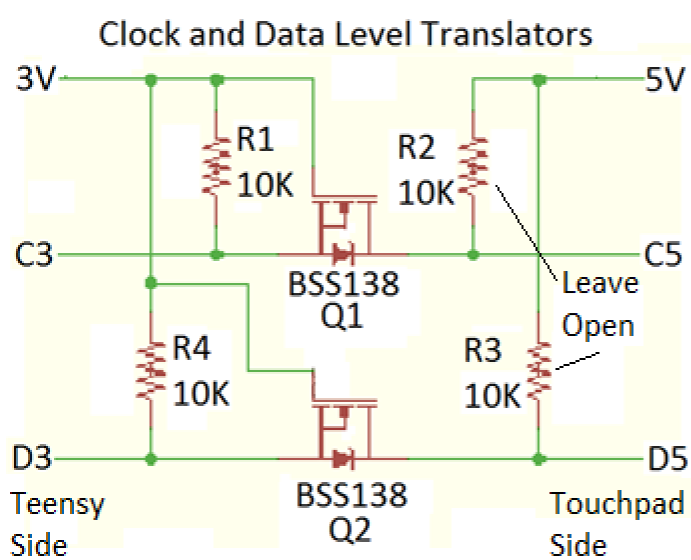
If your touchpad is powered with 3.3 volts, the Teensy LC doesn't need any level translation because the pull up resistors will go to 3.3 volts.

If your touchpad is powered with 5 volts and ohm meter measurements of the clock and data lines show no pullups, there is still a chance that the touchpad chip contains active pullups made from small (weak) PFETs that only come alive when powered. To check for active pullups, power the touchpad with 5 volts. Just measuring the voltage on the clock and data lines won't tell you much because they will probably float to power. A better test is to pull the signal low by temporarily connecting 10K resistors from clock to ground and from data to ground. If the clock and data voltage is near zero volts, there are definitely no pull ups so you won't need a level translator. Add your own 10K pull ups on clock and data to 3.3 volts and the Teensy LC should work fine.

If the clock and data voltage is well above ground when pulled down with a 10K resistor, the touchpad has active pull ups to 5 volts and you will need to level translate the clock and data signals for a Teensy LC. A level translator is also needed if ohm meter measurements showed pullup resistors. There are several ways to level shift the signals: The [Adafruit 757] level converter can be used or the surface mount pads on my Keyboard Scanner LC circuit board shown below can be populated with FETs and resistors to make your own level translator. The 10K resistors on the 5 volt side of the translator should not be installed since the touchpad already has pullups.



Surface Mount pads for
Clock and Data Level Translators



- The 10K 0603 resistors are Digikey part number RHM10KADCT-ND
- The BSS138 FETs are Digikey part number BSS138CT-ND

Another (hack) method to translate a 5 volt signal is to make a voltage divider by adding a pull down resistor that overpowers the small active (PFET) pull up in the touchpad chip. This method would never be a reliable production solution because the voltage at the divider will vary from unit to unit and with temperature and time. The value of the resistor is determined by experimentation. If your touchpad has resistor pull ups to 5 volts, then adding pull down resistors to make 3.3 volt logic signals should be a reliable solution.

The following touchpad examples will show how to hook up and test a touchpad:

Example 1: The touchpad on the Dell D620 and D630 does not have numbered test points. To determine the pinout, I found the laptop's schematic online and used an ohm meter to trace the signals thru the cable to the touchpad connector as shown in the tables below.

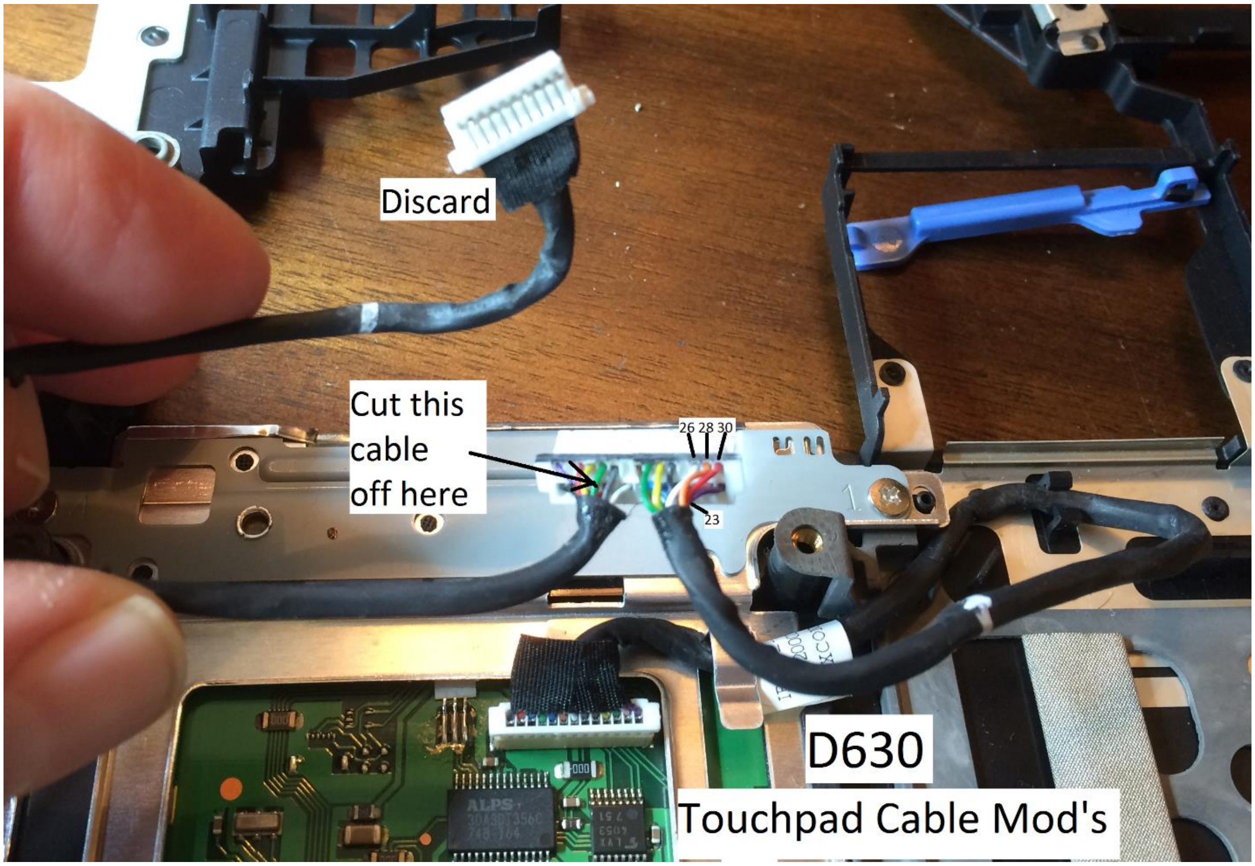
Dell D630
Motherboard Connector

Signal	**Pin #**
Ground	23
Clock	26
Data	28
5 Volts	30

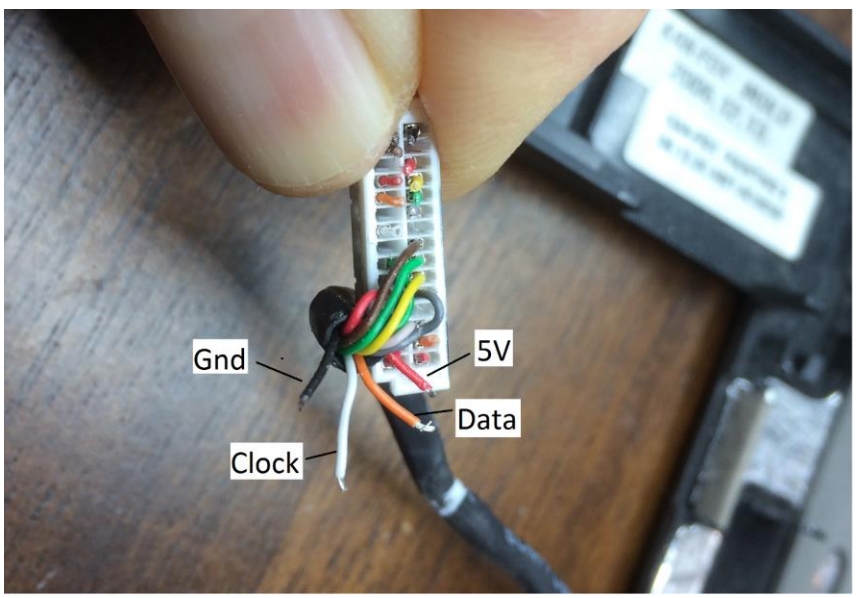
Touchpad Connector

Signal	**Pin #**
Ground	1
5 Volts	2
Data	6
Clock	7

The Dell D620/D630 touchpad cable is a bundle of wires instead of a ribbon cable, making it easy to modify. The picture below shows the cable before I cut off the section I didn't need.

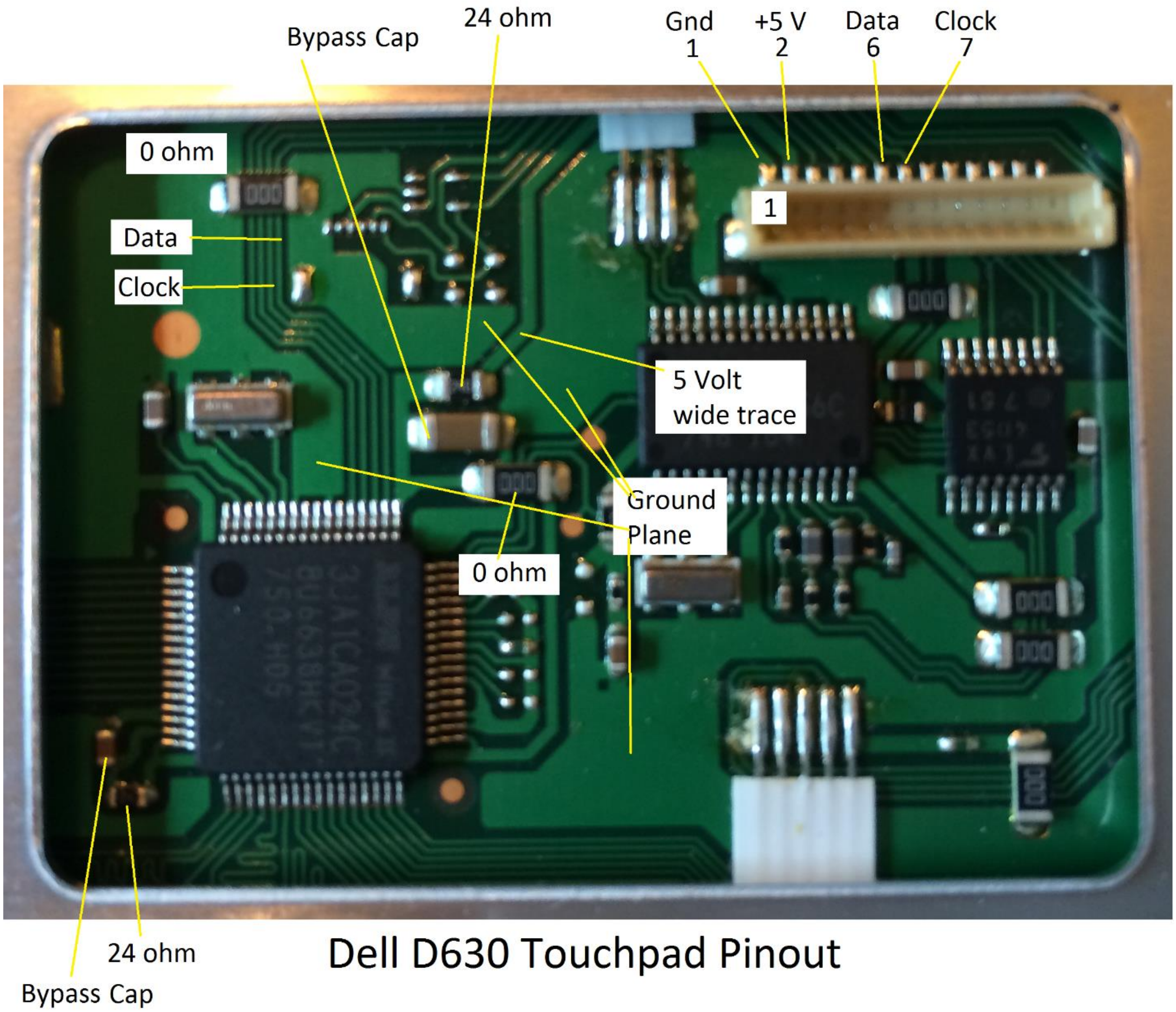


Then I cut the 4 touchpad wires shown below at the motherboard connector so I could solder header pins for easy attachment to the Teensy.



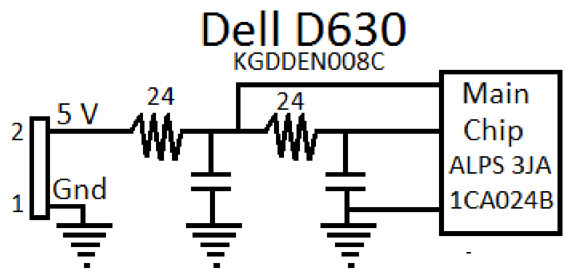
D630 Touchpad Signals

The 5 volt, ground, clock and data locations on the touchpad board are shown below.



Dell D630 Touchpad Pinout

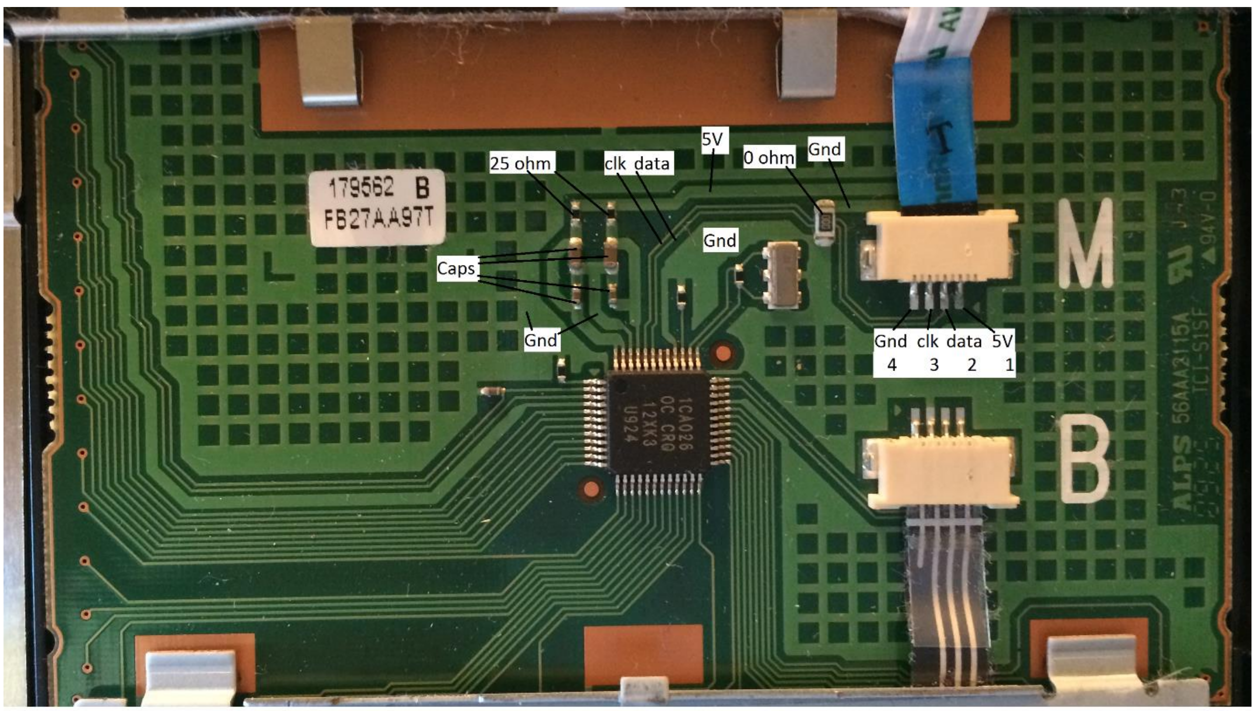
The 5 volts at the touchpad connector passes thru RC filters before connecting to the 3JA1CA024 chip as shown below.



Touchpad RC Power Filter

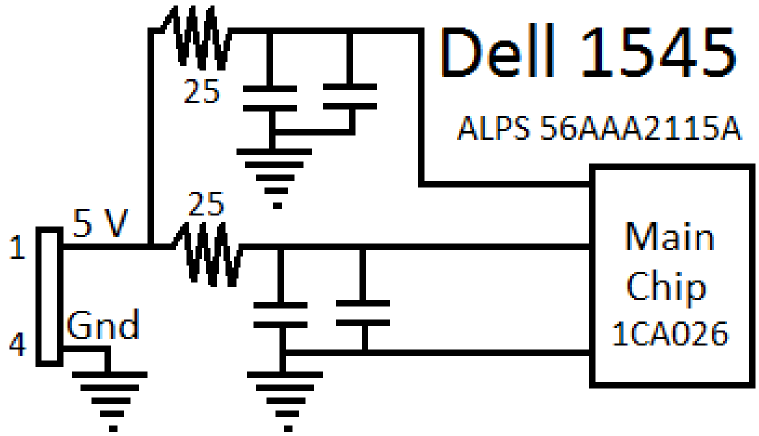
The Clock and Data signals measure 19K Ω to 5 volts and 57K Ω to ground. This pull up/down resistor combination gives an equivalent pull up of 14K Ω to 3.7 volts. No additional pullups are needed to make this touchpad work with a Teensy 3.2. I didn't bother testing the touchpad with a Teensy LC because the keyboard for this laptop has more pins than the LC can handle. The Dell D630 keyboard and touchpad code for the basic KVM and the Pi KVM can be downloaded from my repo.

Example 2: The Dell 1545 touchpad shown below does not have numbered test points and it has a 4 pin connector that is cabled to the motherboard.



Dell 1545 Touchpad Pinout

Ground is the connector pin tied to the large ground plane that covers most of the board. Power is the wide trace that goes to the low value resistors of an RC filter. The other side of the resistors go to capacitors for the filter (see schematic below).

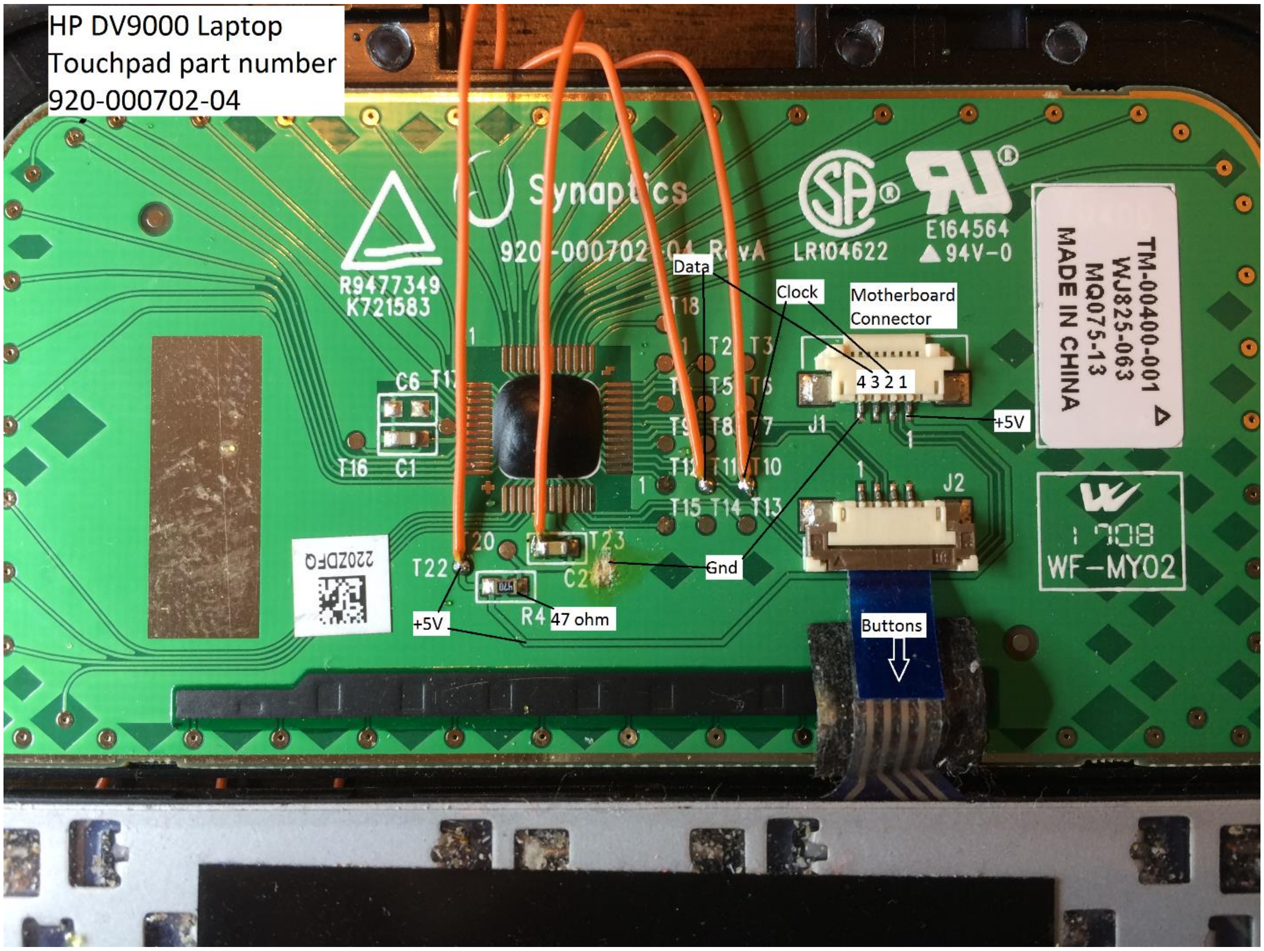


Power RC Filters

The other 2 pins on the connector are the clock and data and they are identified by trial and error. An ohm meter shows both pins measure open to power so no obvious pull ups. To see if the chip contains active pull ups, I connected 10K resistors to ground on the clock and data. Powering up the touchpad with 3.3 volts (just to be safe) showed the clock and data at about 2.5 volts indicating active pull ups. If there were no pull ups, the clock and data would have been at 0 volts. I removed the 10K resistors and connected the clock and data to a Teensy 3.2. I tried swapping the clock and data lines but the touchpad would not work at 3.3 volts. Changing the touchpad power to 5 volts, I was able to find the clock and data connections that worked.

A level translator will allow the Teensy LC to reliably work with this 5 volt touchpad. I successfully used the "hack" method of adding clock and data pull down resistors and found that 4.6K on the data and 4K on the clock pulls these signals slightly below 3.3 volts.

<u>Example 3:</u> The Synaptics 920-000702-04 touchpad is from an HP DV9000 laptop. I soldered PS/2 and power wires from a Teensy 3.2 to the standard T10, T11, T22, and T23 Synaptics test points shown below.



It was hard to heat up the ground plane to solder a wire to pad 23 so I soldered to a nearby capacitor instead. The clock and data pins measure several Mega ohms to power. I found some websites that said this touchpad is powered with 3.3 volts and other sites said 5 volts so to be safe, I powered the touchpad with 3.3 volts first. With 10K pull downs on the clock and data, their voltage was about 2.2 volts, indicating there are active pull ups. Removing the pull downs and trying the Teensy 3.2 showed the touchpad worked at 3.3 volts. Out of curiosity, I powered the touchpad with 5 volts and it also worked so apparently this touchpad is capable of dual voltage operation.

No resistors are needed for the Teensy LC when the touchpad is powered with 3.3 volts.

Basic method for pin identification: If you can't find any information for your touchpad, use resistance and voltage measurements to determine the pinout. Most touchpad circuit boards are routed on the top layer and they use the remaining copper to make a ground plane. The ground pin on the connector will be tied to the ground plane. Different sections of the plane will be tied together with zero ohm resistors that jump over the signal traces. The power connector pin will go to a wide trace that routes to one or two low value resistors for the RC filter. If there are too many possible signal pins on the connector to use the trial and error method, use the following procedure to weed out the obvious pins that can't be clock or data:

Only look at connector pins that go directly to the main touchpad chip. Measure and record the resistance for each of these pins to power and to ground. While taking these measurements, push the left and right touchpad buttons to see if the resistance changes. Put 3.3 volts across the touchpad power and ground pins, then measure and record the voltage on each of the possible pins with a 10K pull down resistor to ground followed by the same resistor pulled up to 3.3 volts. Remove the resistors and push the left and right touchpad buttons to see if the pin voltage changes. Analyze the results of the resistance and voltage measurements as follows:

- Eliminate any pin that measures 2KΩ or less to power or to ground.
- Eliminate any pin that has a significant change in resistance or voltage when a button is pushed.
- Eliminate any pin that measures close to 3.3 volts when connected to a 10K pull down resistor.
- Eliminate any pin that measures close to 0 volts when connected to a 10K pull up resistor.
- Group pins together that have similar measurements in order to test with the Teensy.

Hopefully you have reduced the number of possible clock and data signals and can use the trial and error method to find the correct 2 pin combination. Add 10K pull up resistors from the clock and data pins to the supply voltage. You can always try removing them once you get the touchpad working. If you don't know the touchpad supply voltage, try 3.3 volts first. Connect two of the Teensy I/O pins to the suspected clock and data connector pins. Download the touchpad code below into the Arduino IDE and edit the clock and data pins to match the I/O numbers wired to the touchpad. Under Tools, set the code to the Teensy model you are using and to "Keyboard+Mouse+Joystick". Compile and load the code into the Teensy. If the LED on the Teensy stays off, the touchpad received the reset command and gave the proper response over the clock/data bus. Swipe your finger across the touchpad and push the buttons to confirm proper operation. If the Teensy LED is turned on, the touchpad failed to communicate with the Teensy. Confirm no operation by swiping your finger across the touchpad. Swap the clock and data pins in the Teensy code and try again. If that doesn't work, move the jumpers to the next possible connector pins. Keep trying until every pin combination has been tested. If no success, increase the touchpad supply voltage to 5 volts and try all possible pin combinations again. If you are using a Teensy LC, use a level translator to avoid applying 5 volts to the I/O pins.