Microsoft Intellimouse Classic 3360 Sensor Upgrade Instructions

By Russell ‘bst’ Vint – 30th April 2018

1. Tools and parts:

Tools – These are the tools I used:
Essential:
- Soldering Iron: Recommended with 0.5mm tip. Example: Antex M12: http://www.hobbytronics.co.uk/antex-soldering-iron-m12
- Solder: Lead free, rosin core, 0.8mm: http://www.hobbytronics.co.uk/solder-4m?keyword=solder
- Helping Hands: Get two of these if you want to make the job easier: http://amzn.eu/9PpUPMC
- Solder Sucker: http://www.hobbytronics.co.uk/solder_sucker?keyword=solder%20sucker
- Solder Stand: http://www.hobbytronics.co.uk/soldering-iron-stand?keyword=soldering%20iron%20stand
- Wire: Multicoloured helps, something like this: http://www.hobbytronics.co.uk/ribbon-cable-10core-15ft or this: https://www.ebay.co.uk/itm/322860010747
- T6 Torx Screwdriver (Intellimouse uses torx screws)
- Glue, epoxy, or glue gun
- Blu Tack (for holding the sensor etc in place for testing)
- Rotary tool (eg. Dremel), with cutting disc for PCB, and use the grinding stone and burr tools for the plastic.
- Electrical tape and optionally, heat shrink.

Not essential but recommended:
- Wire strippers: https://www.diy.com/departments/mac-allister-stripper-pliers/677445_BQ.prd
- Tip Tinner: (makes tinning the soldering iron easy): http://amzn.eu/fSKikte
- Tip Cleaner: There is a tip cleaner on the soldering iron stand but it needs to be wet and cleaned every time, this is a bit easier: http://amzn.eu/4XCLIn5

Parts
- Teensy 2.0, make sure its genuine or you could have problems. Just buy it from a reputable store. This is where I got mine: http://www.hobbytronics.co.uk/teensy-20?keyword=teensy%202.0
- Tindie 3360 sensor board: https://www.tindie.com/products/jkicklighter/pmw3360-motion-sensor/
- USB Connector: https://www.ebay.co.uk/itm/162959114305 (10pcs Mini USB B 5 Pins Male Jack Socket Adapter Solder Wire Connector)

Not essential but recommended:
- IE3 Hyperglides: http://www.hyperglide.net/?hg=ms_skates_1
- Ceesa Paracord: http://www.overclock.net/forum/261-artisan-sales-forum/1621381-paracord-mice-cable-made-ceesa.html
- RGB LED: https://www.mouser.co.uk/ProductDetail/604-WP154A43VBDZGWCA
- RGB LED Resistors:
- 68 Ohm: https://www.mouser.co.uk/ProductDetail/270-68-RC
- 82 Ohm: https://www.mouser.co.uk/ProductDetail/270-68-RC
- 100 ohm: https://www.mouser.co.uk/ProductDetail/270-100-RC

NOTE: If you use a different RGB LED to mine, you will need to use different resistors, and it may need negative power (common cathode) instead of positive (common annode). I recommend you use the same LED as I did!

2. Preparing the shell

This stage requires the rotary tool. Its best done outside, as it creates a lot of dust and debris. Wear some eye protection, sometimes hot bits of plastic can fly into your face. It won’t hurt your face but it probably would hurt your eyes.

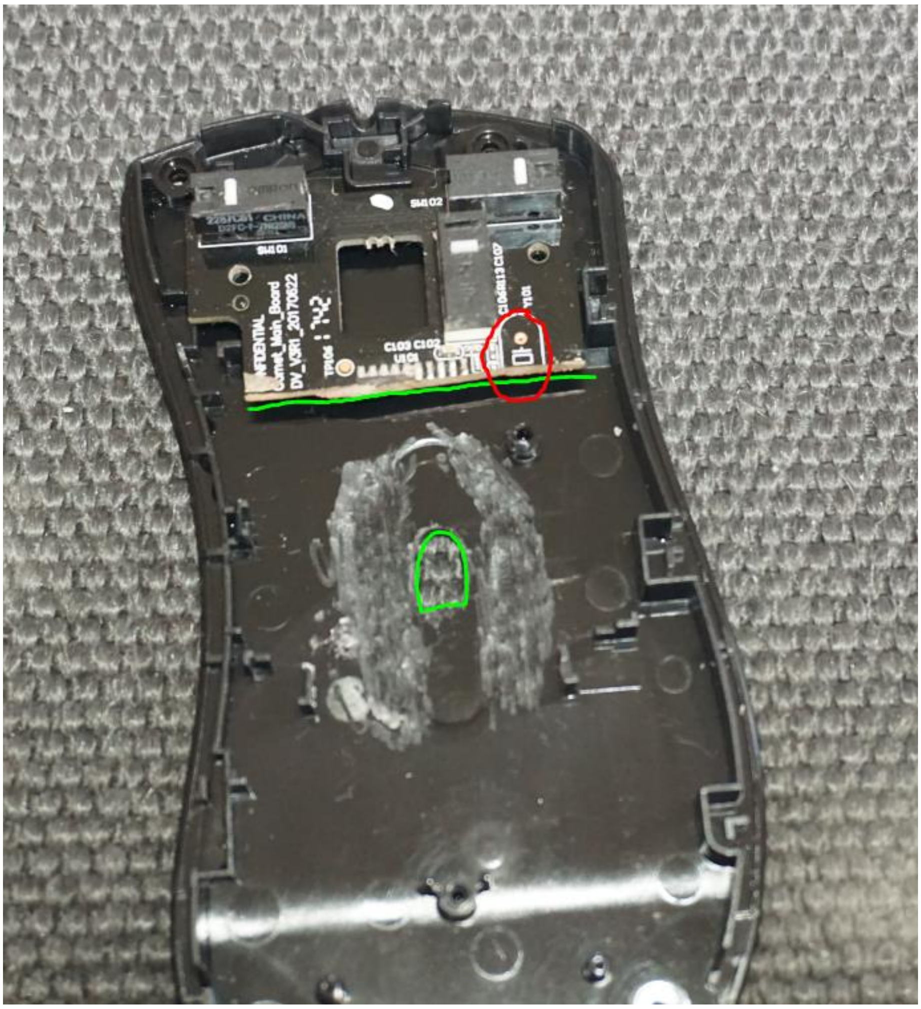
- Start by removing the crystal which is circled in red in the picture (or it'll make it harder to cut through), cover up the microswitches with something to stop the dust getting in them, then cut across the PCB in a straight line.

- Cut the side button ribbon cable from the main PCB, leave as much of the cable as possible, so you can connect it to the teensy. If you cut too much away, it won't be long enough to reach.

- Now prepare the sensor hole area, remove small bits at a time. Keep placing the sensor onto the area to see if it fits, then remove more if necessary. Make sure not to remove the sensor lens at any point, if dust gets into the camera hole, it could ruin the sensor.

- For the base, the sensor lens must sit on it flat. The recess that is already there, is too narrow for the sensor to sit inside, so there are two options:

Option 1 is to widen the recess, this is what I done and it works ok. It also provides a rough surface for the glue to stick to.
Option 2 is to make the lens narrower. I haven't done this before, but it should work. Its something you will have to decide for yourself which method you prefer. If you do this, then when you remove the lens to cut it, cover the sensor camera hole with some tape to stop dust getting inside, and keep it well away from any dusty areas, and make sure you clean the dust from the lens very well after cutting it.

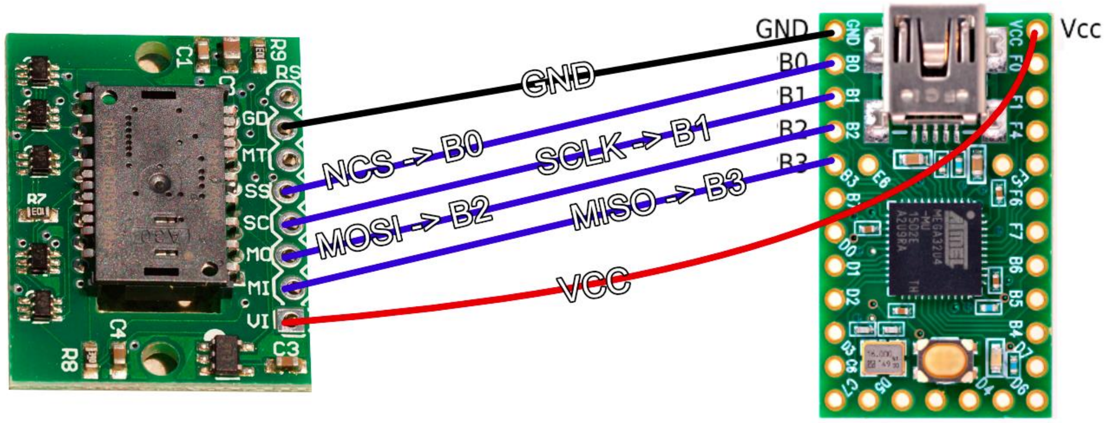


3. Connecting everything to the Teensy

This is fairly simple, but time consuming. There isn't much I can do to help, except show which wires connect where! Just make sure not to cut the wires too short. First I will show the whole list, then show a diagram for each stage.

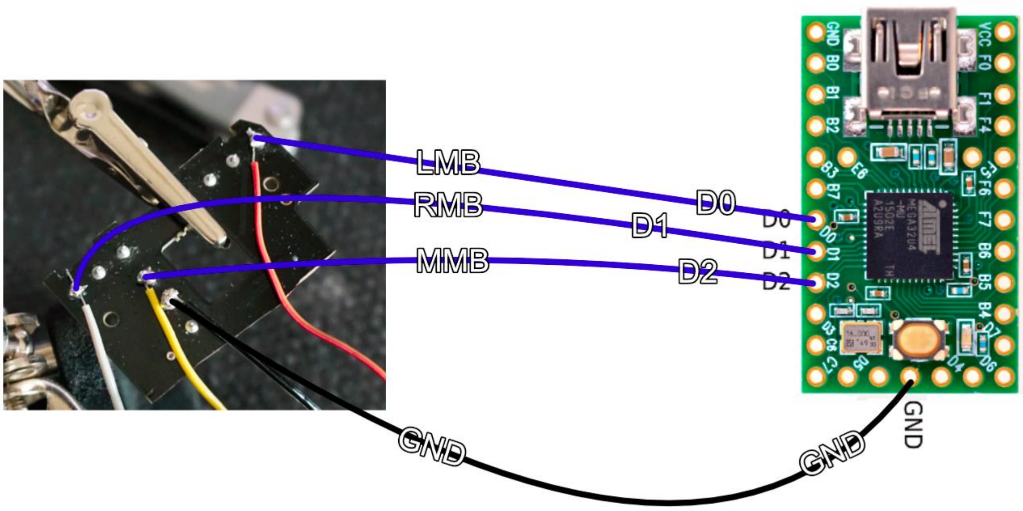
Component Pin	Teensy Pin
3360 Sensor NCS (SS)	B0
3360 Sensor SCLK (SC)	B1
3360 Sensor MOSI (MO)	B2
3360 Sensor MISO (MI)	B3
3360 Sensor VCC (VI)	VCC
3360 Sensor GND (GD)	GND
Left Mouse Button (LMB)	D0
Right Mouse Button (RMB)	D1
Middle Mouse Button (MMB)	D2
Front Buttons Ground (GND)	GND
Rear Side Button	D3
Front Side Button	D4
Wheel B (Scroll Up)	C6
Wheel A (Scroll Down)	C7
Side Buttons & Wheel Ground (GND)	GND
Red LED -> 100 Ohm Resistor ->	B5
Green LED -> 68 Ohm Resistor ->	B6
Blue LED -> 82 Ohm Resistor ->	D7
RGB LED VCC	VCC

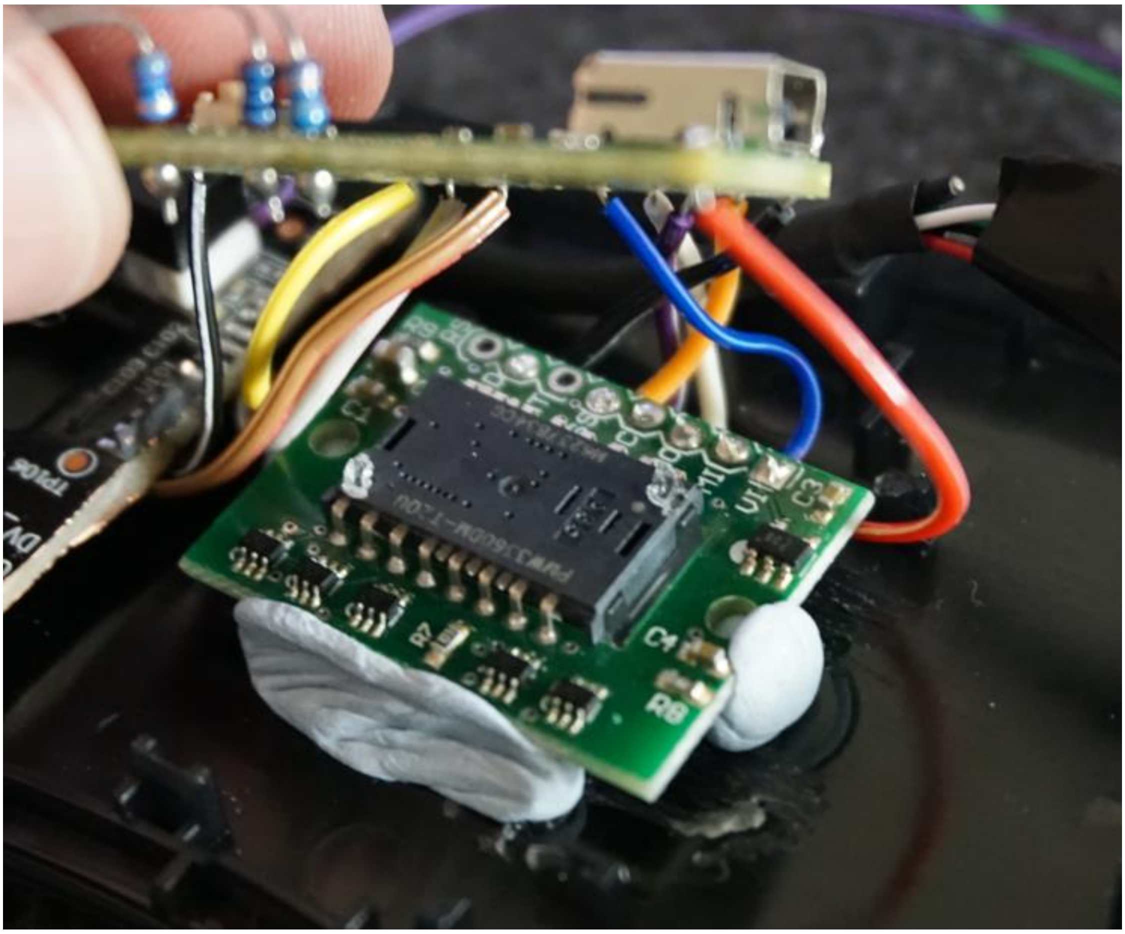
Sensor:



Front Buttons:

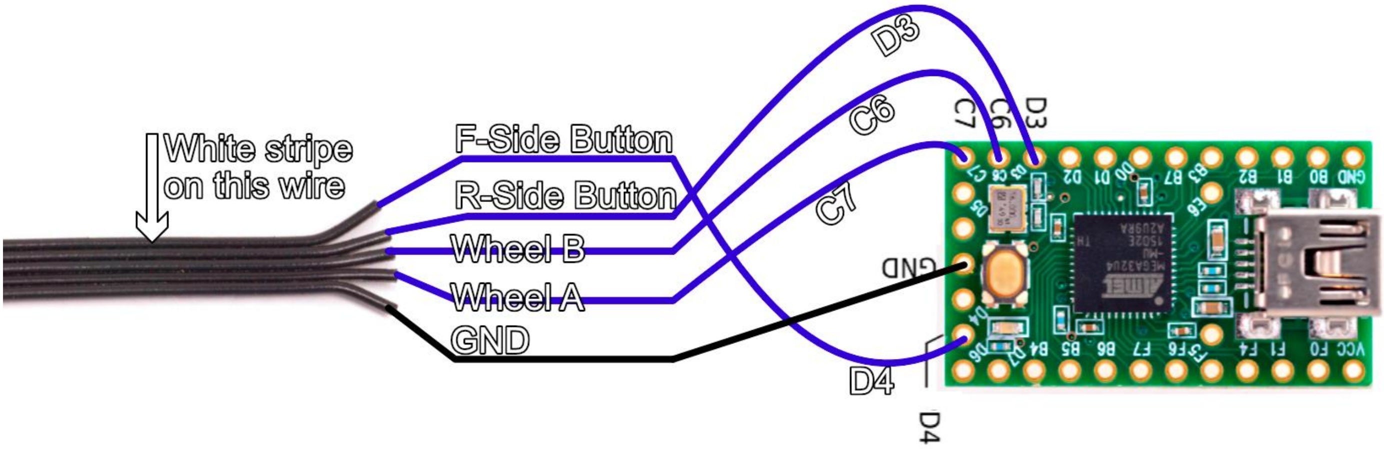
The main thing to remember here is to connect the middle pin of the middle mouse button to GND, that will feed ground to all the buttons.





Side Buttons and Wheel:

This uses the same GND pin on the teensy as the front buttons, but its ok, just twist them both together and solder them in. The ribbon cable has a white stripe on the front side button wire, so use that to know which wire does what.

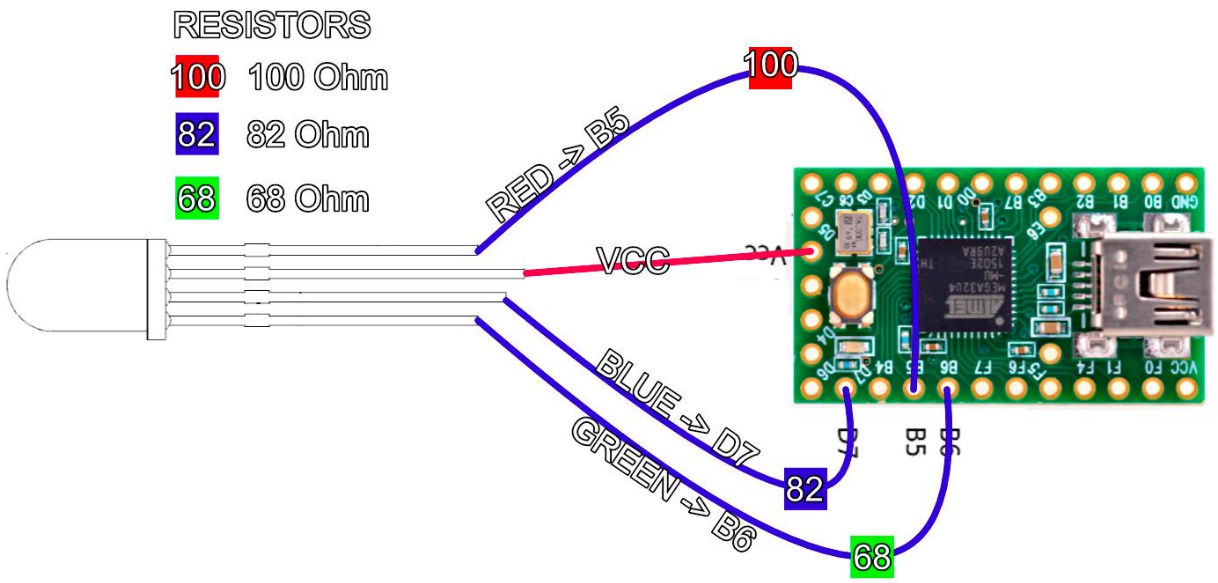


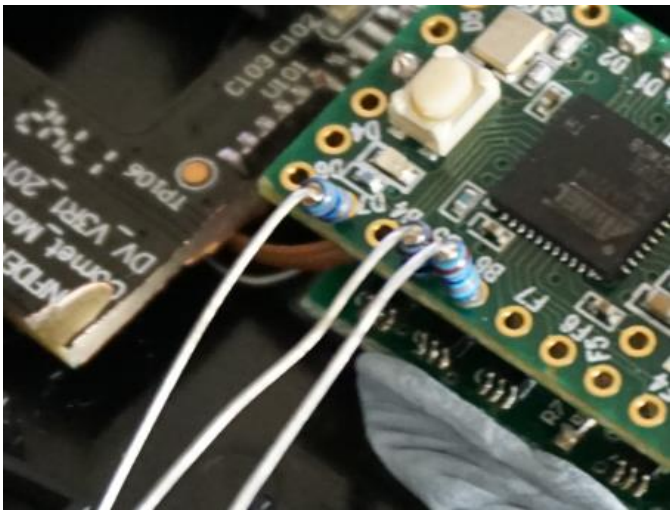
RGB LED

!IMPORTANT! You must connect the resistors in between the LED pins and the Teensy pins.

This is quite fiddly which is why it should be done after the other connections. If you look at the photo below the diagram, I connected the resistors directly to the Teensy pins, but you don't have to do it like that. Make sure to insulate the LED pins and exposed resistor wire with electrical tape or heat shrink. Make sure you have enough length in the wires so the LED can reach over to the stand (where the original white LED used to sit).

Pay close attention to the picture of the LED, it is very precise. The longest pin is the VCC, second longest is blue. Use these as a reference so you connect the wires to the correct pins!

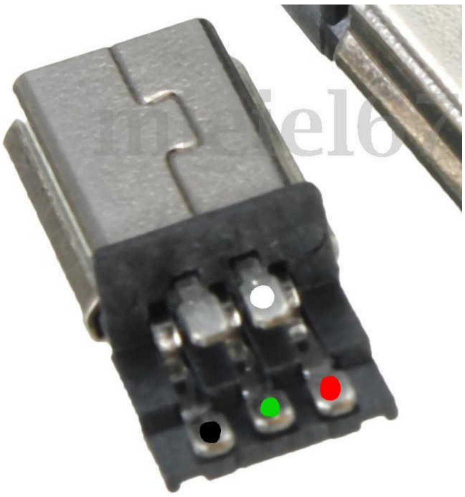




4. Connecting the USB Plug to the USB cable

Now we have to connect the USB cable to the USB connector, so we can plug it into the teensy.

This is very fiddly and if you're new to this, then its best to leave it until after you've soldered all the wires to the teensy, that way you have some practice



All you really have to do is to match the cable colours to the tabs on the USB connector in the picture above. **Make sure NONE of the tabs are touching each other, or that there is solder connecting them together. Don't guess that its ok, make 100% sure, or it can damage things – your PC, the teensy, etc.**

You don't need to worry about connecting the Shield cable, it's a black cable but usually its not shiny. The ground cable is usually a shiny black cable. The Shield cable is also usually thicker (and the last one in the row on the USB PCB plug). The paracord's don't have a shield cable so you don't need to worry about it if you're using a paracord.

5. Connecting the mouse to the PC and loading the firmware

Now, at this point its almost finished. Don't glue the sensor in place yet, instead, you first want to check your work and make sure there are no mistakes, and that there are no short circuits or bridged connections (eg, two pins on the teensy or 3360 sensor touching each other). Also double check you have everything soldered to the correct pins.

Download this program:
https://www.pjrc.com/teensy/loader_xp.html

Read that page so you can get familiar with the program.

Run the program Teensy.exe.

Download the Firmware for this project here:
https://drive.google.com/file/d/1WMrLgY8O2XIBPOJ35OaYfekqFuTDtt8A/view?usp=sharing

Extract the Zip.

Make sure no metal parts of the wires or PCBs are touching each other, then plug in the USB cable into the teensy, then plug the other end into your PC.

The Teensy should be blinking its light. Now, press the button on the Teensy (the tiny button on its PCB). The Teensy.exe program should change, allowing you to press more buttons.

- Drag and drop "bst3360.hex" into the teensy window (onto the picture of the Teensy).

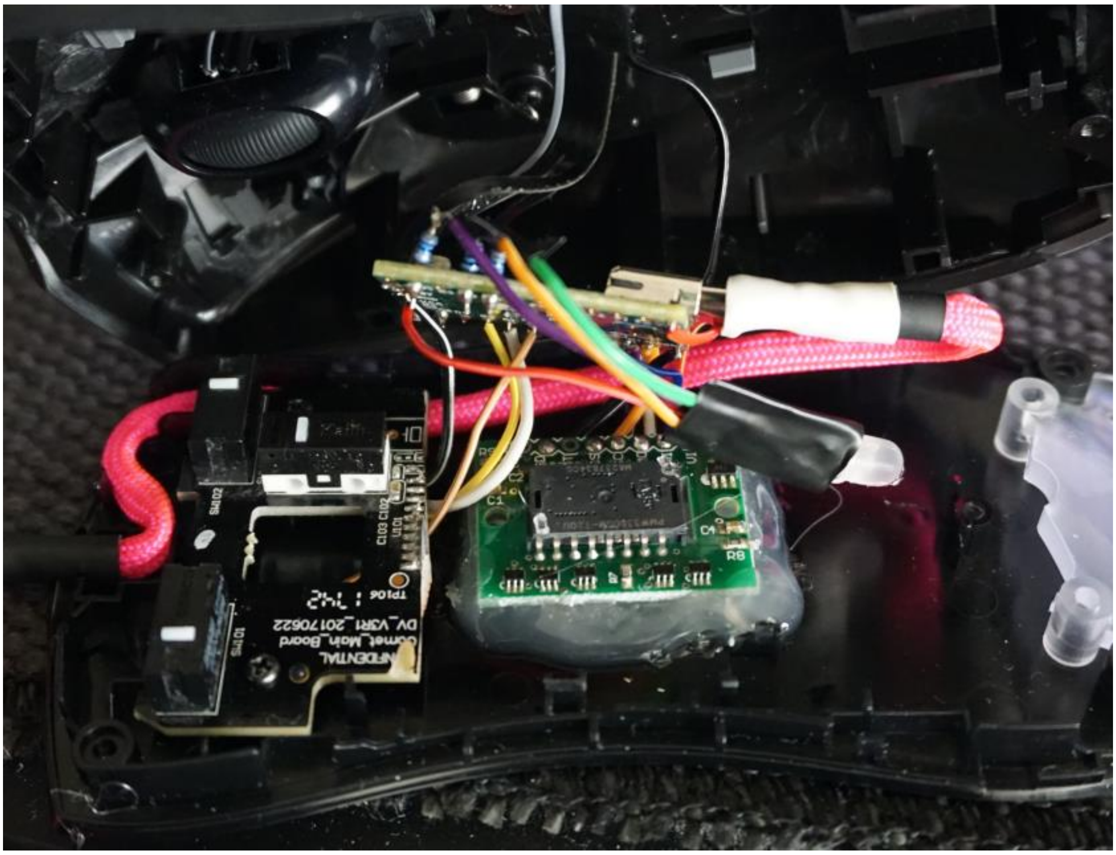
- Now press this button ⤵ to program the firmware onto the mouse.

- When it has finished, press this button to reboot the mouse: ➡

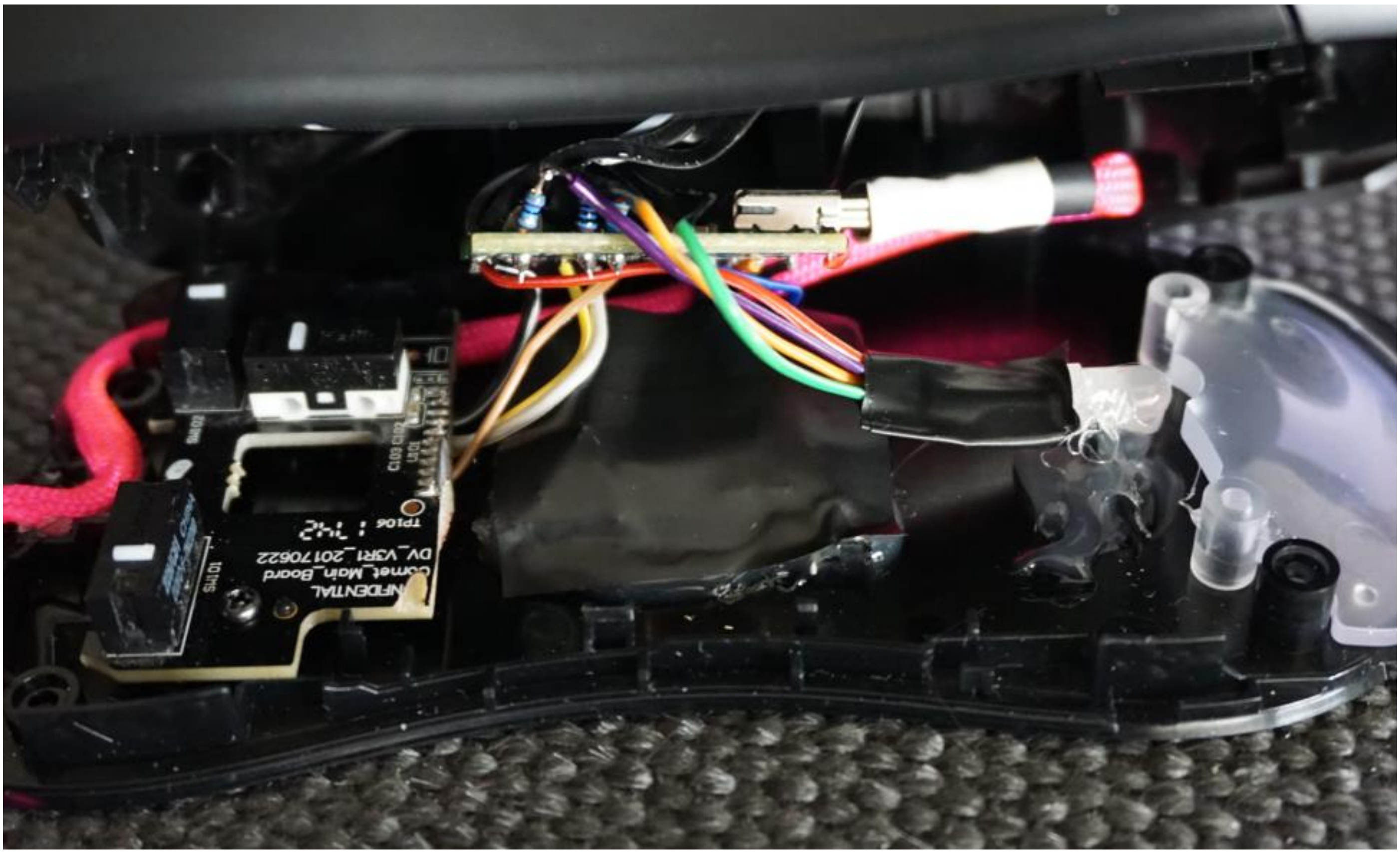
Now you should be able to move the mouse around and click the buttons, scroll the wheel etc. If not, you need to go back and check everything.

6. Finishing up

- Glue the sensor in place with the glue gun, make sure its straight, or at the angle you prefer.



- Make sure to insulate every exposed bit of metal that could come in contact with another. For example you don’t want the pins on the bottom of the teensy somehow touching the 3360 sensor pins, so use some electrical tape over the top of the sensor to help prevent this.



7. Functions of the Firmware

The only thing you can do is to change the profile. By default it runs at 1000hz, has angle snapping off, and the LOD is set to low.

- By default it is set to 800 DPI with a red colour.
- To change to a different profile, hold down one of the side buttons when plugging the mouse in:
---- Hold down the *front side button* for 1600 DPI with a purple colour.
---- Hold down the *rear side button* for 400 DPI with a pink colour.
---- If you don't hold down any buttons when plugging in the mouse, it will be 800 DPI with a red colour.

If you're happy with these default settings, then you don't need to do steps 8 & 9 below (editing and compiling the firmware). You only need to do those steps if you want to change things.

8. Editing the Firmware

You only need to edit Main.c, you don't need to worry about the other files, but those other files must stay in the folder with main.c when you come to compiling the firmware, or it won't compile.

To choose different colours, and change the brightness of the LED, edit this part of main.c (starts at line 252). (Hint: p1 = profile 1):

```
uint8_t p1_led_colour = 1; // Red (DEFAULT PROFILE)
uint8_t p2_led_colour = 2; // Pink
uint8_t p3_led_colour = 4; // Purple
uint8_t p1_led_brightness = 4; // 0 = 0, 1 = 0.25, 2 = 0.5, 3 = 0.75, 4 = 1
uint8_t p2_led_brightness = 4; // 0 = 0, 1 = 0.25, 2 = 0.5, 3 = 0.75, 4 = 1
uint8_t p3_led_brightness = 4; // 0 = 0, 1 = 0.25, 2 = 0.5, 3 = 0.75, 4 = 1
```

The colours and their respective numbers are:

White	0
Red	1
Pink	2
Magenta	3
Violet	4
Blue	5
Sky Blue	6
Cyan	7
Green	8
Toxic Green	9
Yellow	10
Orange	11

To change the DPI is more complicated.
First, find the below part of the code in main.c, its on line 284.

```
uint8_t dpis[] = {3, 7, 15};
```

Enter into the brackets which DPI you want available.
The numbers represent the DPI, for example:

(**3**+1)*100 = **400 DPI**
(**15**+1)*100 = **1600 DPI**

So the below example can be read as: 400, 800, 1600.

```
        uint8_t dpis[] = {3, 7, 15};
```

This one is 1000, 2000, 3000:

```
        uint8_t dpis[] = {9, 19, 29};
```

The **first** number is for profile **2** **(rear side button)**
The **second** number is for profile **1** **(Default profile)**
The **third** number is for profile **3** **(front side button)**

So if you use this setting:

```
        uint8_t dpis[] = {3, 7, 15};
```

Then the DPIs will be assigned like this:

Profile 2: 400 DPI
Profile 1: 800 DPI
Profile 3: 1600 DPI

You can set any DPI value between 200-12,000, in steps of 100. ***You must enter it exactly as shown above, or it won't work!***

To turn angle snapping on, edit this setting on line 276. 0 is off, 1 is on:

```
        uint8_t angle_index = 0;
```

The final thing I will explain is how to change the LOD. You can change other options too, but there isn't much. To find out what the other options are, take a look at the 3360 datasheet, which can be found here:
http://www.pixart.com/upload/PMS0058-PMW3360DM-T2QU-NNDS-R1.30-06042016_20160902201411.pdf

Changing the LOD is pretty easy, just change where it says 0x02 (in bold), to either 0x03, or 0x00. The default setting is 0x02, which is 2mm LOD. 0x03 is 3mm LOD. 0x00 is explained below.

```
        spi_write(0x63, 0x02); // LOD: 0x00 disable lift detection, 0x02 = 2mm, 0x03 = 3mm
```

If you set the LOD to 0x00, then it will be very high. You can fine tune it with these settings:

```
        spi_write(0x2b, 0x10); // Minimum SQUAL for zero motion data (default: 0x10)
        spi_write(0x2c, 0x0a); // Minimum Valid features (reduce SQUAL score) (default: 0x0a)
```

You can play around with those settings and find a good LOD, however, its not easy. I wouldn't bother unless you are really bothered about the LOD and want to spend a long time testing.

9. Compiling the Firmware

When you've finished editing and compile the Firmware, you will get the bst3360.hex file, which you can then load onto the Teensy, the same way you did before. The only difference is you don't need to open the mouse again to press the button on the Teensy. If you hold down the wheel button while plugging the mouse in, it will do the same thing as pressing the teensy button (it will enable the bootloader).

How to compile on Linux

If you're in Linux, you just need to make sure you have the AVR GCC Toolchain (http://maxembedded.com/2015/06/setting-up-avr-gcc-toolchain-on-linux-and-mac-os-x/), right click in the folder where main.c is and open a terminal, then type "make", and it will compile the firmware.

How to compile on Windows

According to Qsxcv from overclock.net, there is a better way of doing it, but this is the way I use and it works fine (I'll quote Qsxcv after it, if you prefer his method):

1) Download Oracle VM Virtualbox from here: https://www.virtualbox.org/
2) Follow instructions here to install Ubuntu: https://askubuntu.com/questions/142549/how-to-install-ubuntu-on-virtualbox
3) Once thats all done, load up Ubuntu and follow these instructions to install the AVR GCC Toolchain: http://maxembedded.com/2015/06/setting-up-avr-gcc-toolchain-on-linux-and-mac-os-x/
4) Now, all you have to do is transfer your code into the shared folder, then, in ubuntu, go to the shared folder (the one that has your main.c etc inside), right click and then click on open terminal in this folder. Once the terminal comes up, just type "make" without the quotes, and it'll compile your code. Then you'll have a hex file which you can use with the teensy loader.

Qsxcv's comment about it:
"on windows, easier to use msys2 and precompiled avr-gcc binaries
will post more details when have time"
http://www.overclock.net/forum/375-mice/1561041-reverse-engineering-3366-a-9.html#post26325150

10. The End

If you completed this project, congratulations! Feel free to post about it with some photos on the youtube video, it'd be interesting to see!

https://www.youtube.com/watch?v=nyb6M89QrWI

11. Disclaimer

I am not responsible for any damage you cause to your property or yourself while following these instructions or using the firmware, these instructions are a rough guide, please take the time to learn from professional sources if you are unsure about anything.