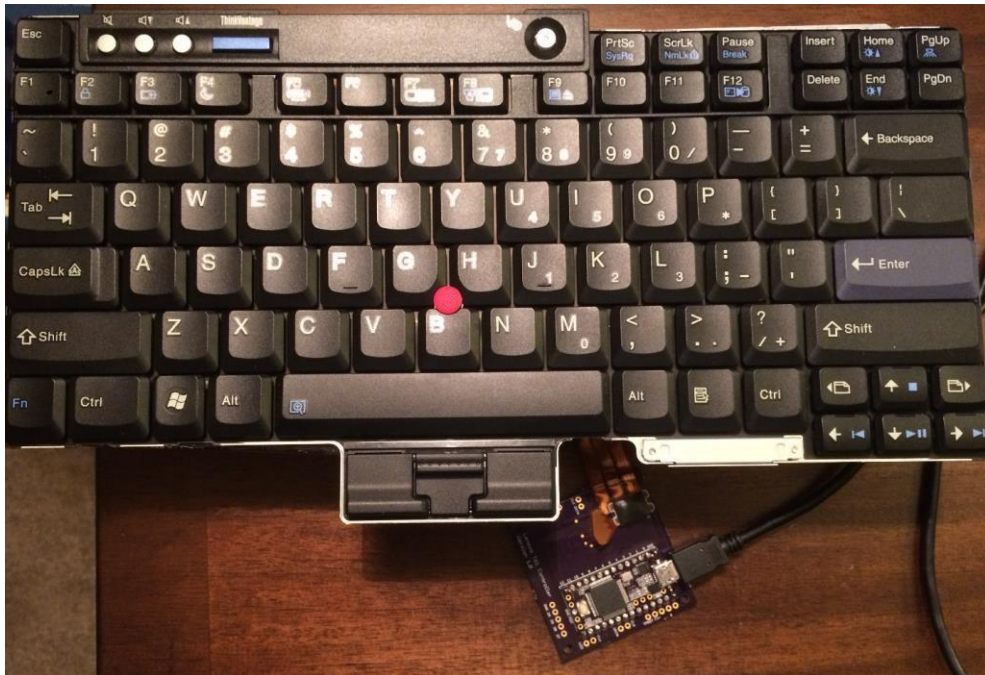
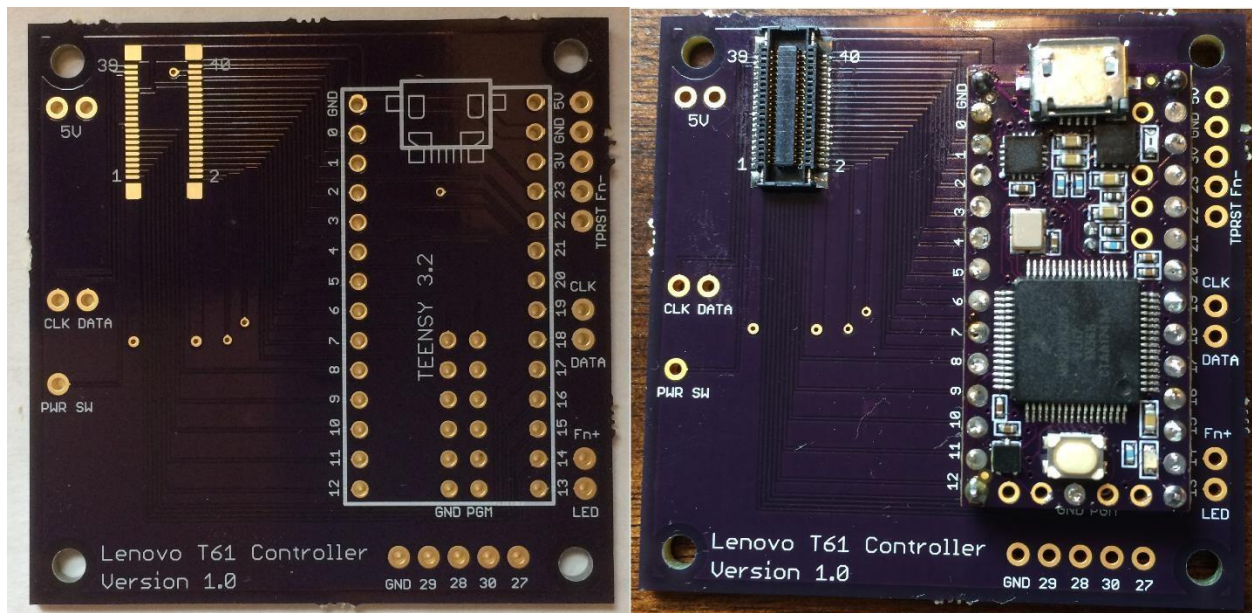


This document will describe how I made a USB keyboard from a Lenovo ThinkPad T61 keyboard using a Teensy 3.2 microcontroller. All associated files are in the Teensy 3p2 folder under the Lenovo_Thinkpad_T61 folder at my [Github repo](#). I have done similar conversions with other laptop keyboards but all of them had flexible printed circuit (FPC) cables that ended with exposed metal traces which fit into a generic connector. Lenovo T60, T61, T400, T500 laptops and their variants have a 44 pin connector plug soldered to the end of the FPC cable. The Lenovo motherboard has the mating socket connector. I designed a circuit board that contains a Teensy 3.2 microcontroller and a 44 pin socket connector. The picture below shows the circuit board connected to a ThinkPad T61 keyboard.



The bare and assembled board is shown below.



Trackpoint PS/2 Pull-Ups 4.7K to 10K

Power Switch No connection to Teensy

Lenovo T61 Controller Version 1.0

Trackpoint 5V

Trackpoint GND

3.3V from Teensy

Fn- Test Point

Trackpoint Reset

Trackpoint Clock

Trackpoint Data

Fn+ Test Point

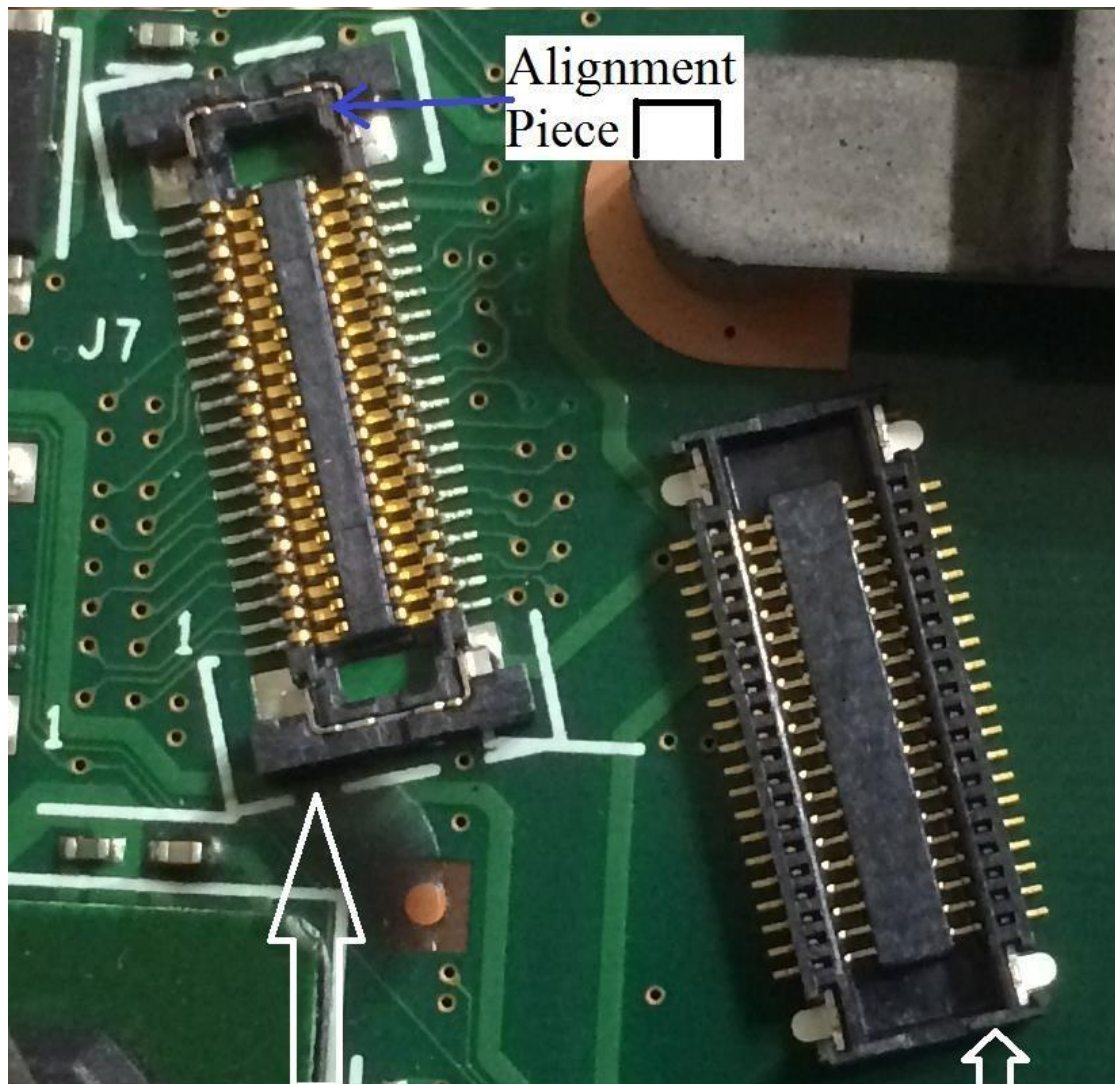
Teensy LED Heartbeat

I/O 30 Scroll Lock

I/O 29 Num Lock

I/O 28 Caps Lock

I/O 27 Loop Sync



Original Connector
JAE AA01B_S040VA1

Substitute Connector
WM6787CT-ND

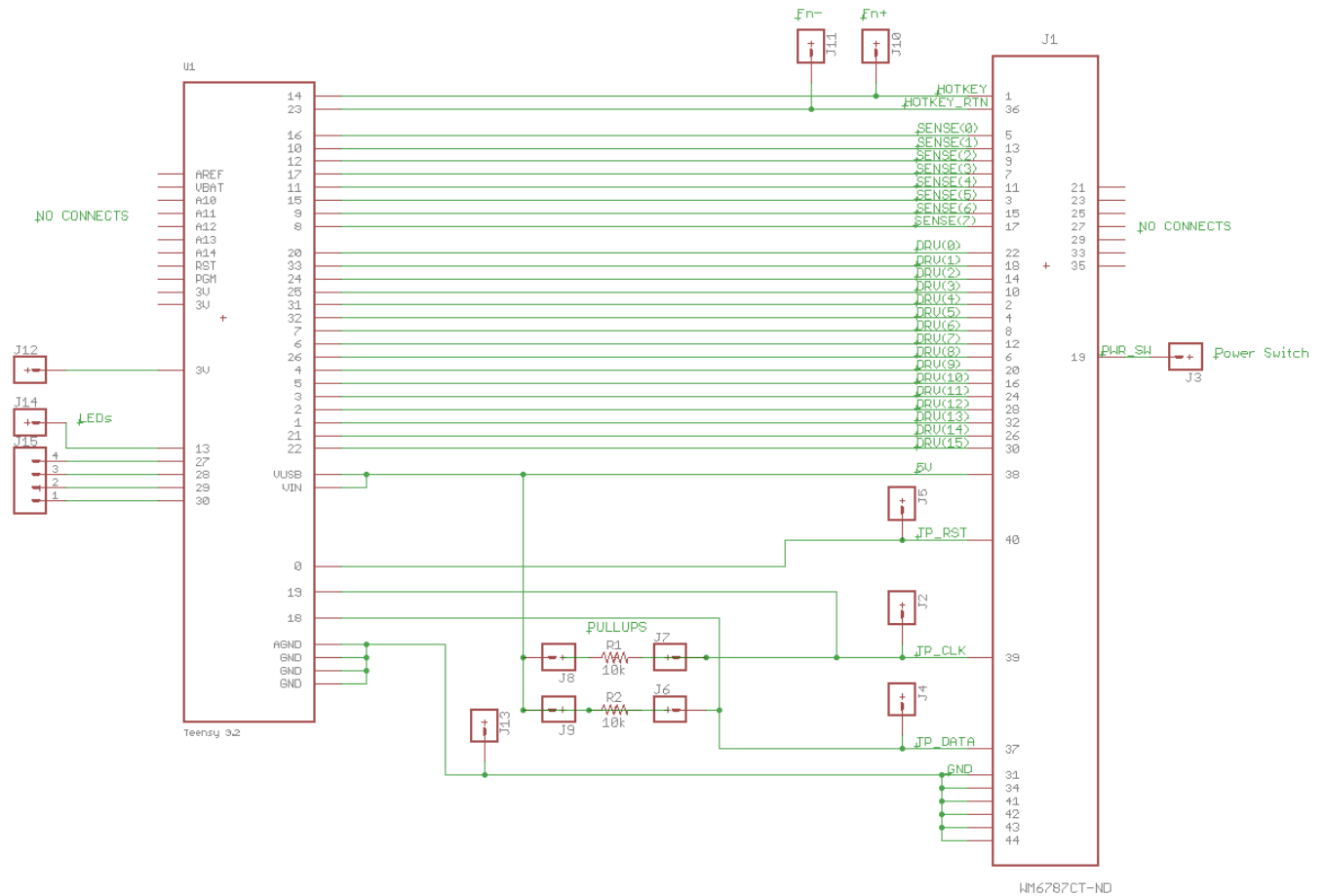
The WM6787CT-ND connector from Digikey will work with the Lenovo T61 keyboard but I did have some trouble with alignment. It took me several attempts to figure out how to center the plug before pushing it into the socket. With the Fn key pressed, I used an ohm meter on the Fn+ and Fn- pads on the board to see if the connector was seated properly. Note that the “on” resistance of the Fn switch is about 225 ohms which may be too high to “beep” your meter. A possible cause of the alignment issue is shown in the picture above. The ends of the Digikey connector are missing the alignment piece found in the original JAE AA01B_S040VA1 connector. Once I got the plug correctly installed in the socket, it worked fine so I don’t view this as a big problem. The alternative to using the Digikey connector is to search for the original JAE AA01B_S040VA1 connector or unsolder it from a broken motherboard. Forum posters have reported that the original connector is hard to find and only available in large lot sizes.

The keyboard connections to the Teensy 3.2 pins are shown below.

Lenovo ThinkPad T61 FPC Connector	Teensy 3.2 I/O number	T61 Schematic Signal Name
1	14	HOTKEY
2	31	DRV<4>
3	15	SENSE<5>
4	32	DRV<5>
5	16	SENSE<0>
6	26	DRV<8>
7	17	SENSE<3>
8	7	DRV<6>
9	12	SENSE<2>
10	25	DRV<3>
11	11	SENSE<4>
12	6	DRV<7>
13	10	SENSE<1>
14	24	DRV<2>
15	9	SENSE<6>
16	5	DRV<10>
17	8	SENSE<7>
18	33	DRV<1>
19	PWR SW Pad	PWR SW
20	4	DRV<9>
21	No connect	NC
22	20	DRV<0>
23	No connect	NC
24	3	DRV<11>
25	No connect	KBDID0
26	21	DRV<14>
27	No connect	KBDID1
28	2	DRV<12>
29	No connect	KBDID2
30	22	DRV<15>
31	Teensy GND	KBDID RTN
32	1	DRV<13>
33	No connect	NC
34	Teensy GND	PWR SW RTN
35	No connect	NC

36	23	HOTKEY RTN
37	18	TP_DATA
38	Teensy 5V	TP_5V
39	19	TP_CLK
40	0	TP_RESET
41 thru 44	Teensy GND	GND

The schematic for the Teensy 3.2 to FPC connector board is given below.



It was easy to find a Lenovo motherboard schematic online that provided me with the connector pins for the DRV and SENSE signals. The Matrix_Decoder_T61.ino code at my repo scans the DRV and SENSE pins and sends key press connection characters over USB for capture by an editor. The key press results in the file Lenovo_T61_io_list.txt at my repo were used to build the matrix table shown below. The Teensy inputs aka Sense<0> thru <7> are the columns across the top and the Teensy outputs aka Drive<0> thru <15> are the rows on the side.

Matrix for the Lenovo ThinkPad T61 – Keyboard Part Number 42T3177

Teensy 3.2 I/O Number	Sense<0> I/O 16	Sense<1> I/O 10	Sense<2> I/O 12	Sense<3> I/O 17	Sense<4> I/O 11	Sense<5> I/O 15	Sense<6> > I/O 9	Sense<7> I/O 8
Drive<0> I/O 20	Back-Tick	1	Q	Tab	A	Esc	Z	
Drive<1> I/O 33	F1	2	W	CapsLock	S		X	
Drive<2> I/O 24	F2	3	E	F3	D	F4	C	
Drive<3> I/O 25	5	4	R	T	F	G	V	B
Drive<4> I/O 31	6	7	U	Y	J	H	M	N
Drive<5> I/O 32	Equal	8	I	Right- Brace	K	F6	Comma	
Drive<6> I/O 7	F8	9	O	F7	L		Period	
Drive<7> I/O 6	Minus	0	P	Left- Brace	Semi-colon	Quote		Forward- Slash
Drive<8> I/O 26	F9	F10		Back- Space	Back-Slash	F5	Enter	Space
Drive<9> I/O 4	Insert	F12			Possible Fn location			Arrow- Right
Drive<10> I/O 5	Delete	F11	VolumeUp	VolumeDown	Mute	Think- Vantage		ArrowDown
Drive<11> I/O 3	Page-Up	Page- Down	GUI		Menu		PageLeft	Page- Right
Drive<12> I/O 2	Home	End				Arrow-Up	Pause	Arrow- Left
Drive<13> I/O 1		Print- Screen	ScrollLock			Alt-L		Alt-R
Drive<14> I/O 21				Shift-L			Shift-R	
Drive<15> I/O 22	Cntrl-L						Cntrl_R	

The Fn “Hotkey” has its own dedicated pins and is not part of the matrix. Fn is wired to connector pins 1 and 36 (Teensy I/O’s 14 and 23). To read the Fn key, I/O 23 is driven low and I/O 14 is read as an input with a pullup. Instead of just grounding the Fn return signal at the connector, I brought both sides of the Fn switch to the Teensy and to pads on the board. This allowed me to temporarily jumper the Fn key into the matrix to confirm I could scan it along with all the other keys. The location I tested in the matrix for the Fn key is the empty cell between I/O 4 and I/O 11 as shown above. Placing the Fn switch in the matrix could be useful if you are trying to save Teensy I/O pins (see Teensy LC controller board). For the Teensy 3.2 implementation, I have kept the Fn key out of the matrix.

With the key matrix information shown above, you can load the TMK software into the Teensy. I have written a keyboard and trackpoint routine as a simple alternative called [Lenovo T61 KBandTP.ino](#). It uses the Teensyduino "Micro-Manager Method" to send keys over USB. All of the normal and modifier keys are supported but only the volume control multimedia keys are part of this routine. A description of the Teensyduino keyboard functions is at www.pjrc.com/teensy/td_keyboard.html.

The PS/2 code for the Trackpoint was coded based on the timing diagrams and uses the PJRC Mouse functions to send over USB. A watchdog timer in the "while loops" won't let the code hang if the Teensy misses a clock edge. This can happen when the Teensy is interrupted by the USB port to receive the keyboard_leds (CAPS LOCK) data.

Trackpoint testing with the Teensy shows that a stronger pullup for the PS/2 clock and data is needed than the weak pullups on the Teensy inputs. My board has thru-hole pads for clock and data pull up resistors to the left of the Digikey connector. 4.7K pullups per the Lenovo schematic worked and 10K's also worked. The [BSS138](#) level shifter board from Adafruit uses 10K pullups and it was tested to make sure it worked for possible use by a Teensy LC. The Teensy 3.2 doesn't need level translators since it has 5 volt tolerant I/O.

I designed a second T61 controller board that uses a Teensy LC based on the information learned from the 3.2 board. To use a Teensy LC instead of a 3.2, see the project description and all associated files in the [Teensy LC folder](#) at my GitHub repo.