Instructions for modifying the USB keyboard code

These instructions will modify the USB keyboard code for a new keyboard. If you are using a Teensy 2.0++, change all “int” in the code to “unsigned int” or it will give compilation errors. The other Teensy’s will compile with or without “unsigned int” for all integer variables.

If using the manual method described in my Instructable, jump to page 4 for instructions. If you are using Marcel’s program, its output will automatically provide the information that needs to be added to the keyboard code. The Python program results are given in a terminal window which should be copied to a text file for transfer to your PC. An example output from Marcel’s program (with added notes) is given below.

```
-------------------------------------------------------
Results:
-------------------------------------------------------

FPC PINS:

8 input pins:
[18, 19, 20, 21, 22, 23, 24, 25]

17 output pins:
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]
-------------------------------------------------------
TEENSY PINS (these have to be copied to the arduino file):
8 input pins:
[8, 16, 9, 15, 10, 14, 11, 26]
17 output pins:
[23, 0, 22, 1, 24, 2, 21, 3, 25, 4, 20, 5, 19, 6, 18, 7, 17]
-------------------------------------------------------
```

Keyboard FPC Input and Output pins

cols_max=8

FPC pins translated to Teensy I/O pins

rows_max=17

In the USB Keyboard code, look for the line:

```
const byte cols_max = ;
```

For this example it should be set to 8;

Look for the line:

```
const byte rows_max = ;
```

For this example it should be set to 17;

Look for the line:

```
unsigned int Col_IO[cols_max] = {     }; // unsigned is only required for 2.0++
```

For this example it should list pins 8,16,9,15,10,14,11,26 inside the curly brackets

Look for the line:

```
unsigned int Row_IO[rows_max] = {     }; // unsigned is only required for 2.0++
```

For this example it should list pins 23,0,22,1,24,2,21,3,25,4,20,5,19,6,18,7,17 inside the curly brackets

The normal, modifier, media, and old_key matrixes are provided by Marcel's Python program and should be copied and pasted over the top of the existing array values as shown in the following screen captures.

Normal Keys in a row column matrix

```
-----------------------------------------------------
KEY copy into int normal[rows_max][cols_max]=
-----------------------------------------------------
{
{0,KEY_INSERT,0,KEY_F12,0,0,0,KEY_RIGHT},
{0,KEY_DELETE,0,KEY_F11,0,0,0,KEY_DOWN},
{KEY_UP,KEY_HOME,KEY_MENU,KEY_END,0,0,0,KEY_LEFT},
{0,KEY_F8,KEY_F7,KEY_9,KEY_O,KEY_L,KEY_PERIOD,0},
{KEY_QUOTE,KEY_MINUS,KEY_LEFT_BRACE,KEY_0,KEY_P,KEY_SEMICOLON,0,KEY_SLASH},
{KEY_F6,KEY_EQUAL,KEY_RIGHT_BRACE,KEY_8,KEY_I,KEY_K,KEY_COMMA,0},
{KEY_H,KEY_6,KEY_Y,KEY_7,KEY_U,KEY_J,KEY_M,KEY_N},
{KEY_F5,KEY_F9,KEY_BACKSPACE,KEY_F10,0,KEY_BACKSLASH,KEY_ENTER,KEY_SPACE},
{KEY_G,KEY_5,KEY_T,KEY_4,KEY_R,KEY_F,KEY_V,KEY_B},
{KEY_F4,KEY_F2,KEY_F3,KEY_3,KEY_E,KEY_D,KEY_C,0},
{0,KEY_F1,KEY_CAPS_LOCK,KEY_2,KEY_W,KEY_S,KEY_X,0},
{KEY_ESC,KEY_TILDE,KEY_TAB,KEY_1,KEY_Q,KEY_A,KEY_Z,0},
{0,0,0,KEY_PRINTSCREEN,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,KEY_PAGE_UP,KEY_PAGE_DOWN,0,0},
{0,0,0,0,0,0,0,0},
}
```

Modifier Keys in a row column matrix

```
-----------------------------------------------------
MODIFIER Copy to int modifier[rows_max][cols_max]=
-----------------------------------------------------
{
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{MODIFIERKEY_LEFT_ALT,0,0,0,0,0,0,MODIFIERKEY_RIGHT_ALT},
{0,0,MODIFIERKEY_LEFT_SHIFT,0,0,0,MODIFIERKEY_RIGHT_SHIFT,0},
{0,MODIFIERKEY_LEFT_CTRL,0,0,0,0,MODIFIERKEY_RIGHT_CTRL,0},
{0,0,0,MODIFIERKEY_GUI,0,0,0,0},
{0,0,0,0,0,MODIFIERKEY_FN,0,0},
}
```

Media Fn keys in a row column matrix

FN Copy to int media[rows_max][cols_max]=

```
{
{0,0,0,KEY_MEDIA_NEXT_TRACK,0,0,0,0},
{0,0,0,KEY_MEDIA_PLAY_PAUSE,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,KEY_MEDIA_VOLUME_DEC,KEY_MEDIA_MUTE,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,KEY_MEDIA_VOLUME_INC,0,KEY_MEDIA_PREV_TRACK,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,KEY_MEDIA_EJECT,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0},
}
```

old_key matrix copy to

ONE boolean old_key[rows_max][cols_max]=

```
{
{1,1,1,1,1,1,1,1},
{1,1,1,1,1,1,1,1},
{1,1,1,1,1,1,1,1},
{1,1,1,1,1,1,1,1},
{1,1,1,1,1,1,1,1},
{1,1,1,1,1,1,1,1},
{1,1,1,1,1,1,1,1},
{1,1,1,1,1,1,1,1},
{1,1,1,1,1,1,1,1},
{1,1,1,1,1,1,1,1},
{1,1,1,1,1,1,1,1},
{1,1,1,1,1,1,1,1},
{1,1,1,1,1,1,1,1},
{1,1,1,1,1,1,1,1},
{1,1,1,1,1,1,1,1},
}
```

The manual method to modify the USB keyboard code is given below. This procedure assumes you have already run the keyboard decoder code which created a list of pins for every key. Then you manually made a key matrix table filled with all the key names as shown in the Instructable.

Const byte rows_max =

- Set this to the number of rows in your matrix

Const byte cols_max =

- Set this to the number of columns in your matrix

int normal[rows_max][cols_max] = { // unsigned int is only required for Teensy 2.0++

- This array should have cols_max items on each line and have rows_max lines.
- Transfer every normal key from your matrix table to this array.
- This array is only for the normal keys, not for Control, Alt, Shift, GUI, Fn, or any Media keys.
- If your matrix table has no normal key in a cell then put a 0 in the array.
- Put a 0 in the cell if your matrix has Control, Alt, Shift, GUI, or Fn keys listed at this location.
- The names given for each key must be as shown in the "All Key Codes" table at: www.pjrc.com/teensy/td_keyboard.html the exception is KEY_MENU, which is not listed on the PJRC table but it works.
- If your keyboard has a key name that does not exist in the PJRC table, it can't be used.
- The PJRC table uses Tilde for the back tick ` key (also known as grave accent key).

int modifier[rows_max][cols_max] = { // unsigned int is only required for Teensy 2.0++

- This array should have cols_max items on each line and have rows_max lines.
- Transfer every modifier key from your matrix table to this array.
- If your matrix table has a normal key or no key listed in the cell, put a 0 in this position.
- The names for the modifier keys are as listed in the PJRC table except the "lefts" listed below: MODIFIER_LEFT_CTRL, MODIFIER_LEFT_SHIFT, & MODIFIER_LEFT_ALT are missing from the PJRC table but they work.
- MODIFIER_FN has been defined at the top of the code so it can watch for an Fn key press for multimedia items. The Fn key by itself is not sent over USB.

Int media[rows_max][cols_max] = {

- This array should have cols_max items on each line and have rows_max lines.
- This table is for the media keys and any other key that are accessed by holding down the Fn key.
- You can only use items that are listed in the PJRC "All Key Codes" table for the Normal, Media Player, and System Control Keys.
- Put a 0 in the matrix if the key has no Fn function or if the function is not supported by PJRC.
- Per PJRC, the media keys are sent using Keyboard.press & Keyboard.release.

boolean old_key[rows_max][cols_max] = {

- This array should have cols_max ones on each line and have rows_max lines.

int Row_IO[rows_max] = { }; // unsigned int is required for Teensy 2.0++

- Use the translation tables on the next page to convert each of the FPC pin numbers to Teensy I/O numbers starting from the first row in your matrix table down to the last row.

int Col_IO[cols_max] = { }; // unsigned int is required for Teensy 2.0++

- Use the translation table on the next page to convert each of the FPC pin numbers to Teensy I/O numbers starting from the first column in your matrix table to the last column.

LED on the Teensy 4.0

For the Teensy 4.0, if your keyboard has a 34 pin FPC cable, you need to unsolder the LED on the Teensy so it doesn't interfere. Comment out any code (if it exists) that was for driving the LED as a CAPS LOCK indicator. Typical CAPs LOCK code to comment out is shown below:

```
#define CAPS_LED 13 // Teensy LED shows Caps-Lock.

if (keyboard_leds & 1<<1) {  // mask off all bits but D1. Test if set
go_1(CAPS_LED); // turn on the LED
}
else {
go_0(CAPS_LED); // turn off the LED
}
```

These are the connector board translation tables for the obsolete Teensy LC and Teensy 3.2.

Teensy LC

FPC Pin #	Teensy LC I/O #
1	23
2	0
3	22
4	1
5	24
6	2
7	21
8	3
9	25
10	4
11	20
12	5
13	19
14	6
15	18
16	7
17	17
18	8
19	16
20	9
21	15
22	10
23	14
24	11
25	26
26	12

Teensy 3.2

FPC Pin #	Teensy 3.2 I/O #
1	23
2	0
3	22
4	1
5	21
6	2
7	20
8	3
9	19
10	4
11	18
12	5
13	17
14	6
15	24
16	7
17	25
18	8
19	33
20	9
21	26
22	10
23	27
24	11
25	28
26	12
27	32
28	31
29	30
30	29
31	16
32	15
33	14
34	LED 13

The LED on the Teensy LC is on I/O #13 which is not used by the FPC connector board.

If using a 34 pin FPC cable with a Teensy 3.2, you must unsolder the onboard LED to free up I/O #13.

These are the translation tables for the Teensy 4.0, Teensy 4.1, and Teensy 2.0++ connector boards.

Teensy 4.0

FPC Pin #	Teensy I/O #
1	23
2	0
3	22
4	1
5	21
6	2
7	20
8	3
9	19
10	4
11	18
12	5
13	17
14	6
15	29
16	7
17	31
18	8
19	33
20	9
21	32
22	10
23	30
24	11
25	28
26	12
27	27
28	26
29	25
30	24
31	16
32	15
33	14
34	13 LED

Teensy 4.1

FPC Pin #	Teensy I/O #
1	23
2	0
3	22
4	1
5	21
6	2
7	20
8	3
9	19
10	4
11	18
12	5
13	17
14	6
15	16
16	7
17	15
18	8
19	14
20	9
21	10
22	11
23	12
24	24
25	25
26	26
27	27
28	28
29	29
30	30
31	31
32	32
33	33
34	41

Teensy 2.0++

FPC Pin #	Teensy I/O Pin (Silkscreen)	Teensy I/O number
1	B7	27
2	B6	26
3	D0	0
4	B5	25
5	D1	1
6	B4	24
7	D2	2
8	B3	23
9	D3	3
10	B2	22
11	D4	4
12	B1	21
13	D5	5
14	B0	20
15	A0	28
16	E7	19
17	D7	7
18	E6	18
19	E0	8
20	E1	9
21	F0	38
22	C0	10
23	F1	39
24	C1	11
25	F2	40
26	C2	12
27	F3	41
28	C3	13
29	F4	42
30	C4	14
31	F5	43
32	C5	15
33	F6	44
34	C6	16
35	F7	45
36	C7	17

If using a 34 pin FPC cable with a Teensy 4.0, you must unsolder the onboard LED to free up I/O #13.

The onboard LED is not used by the keyboard connector for the Teensy 4.1 and Teensy 2.0++ connector boards. Many of the example keyboard routines at my GitHub repository will use the onboard LED to display CAPS Lock.