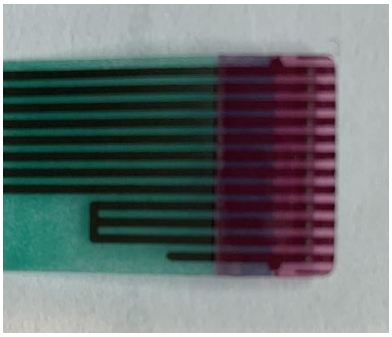
Fujitsu Lifebook 755Tx laptop

Other numbers on backside sticker: FPC95-0253-01 FPC07005A CA04259-B212

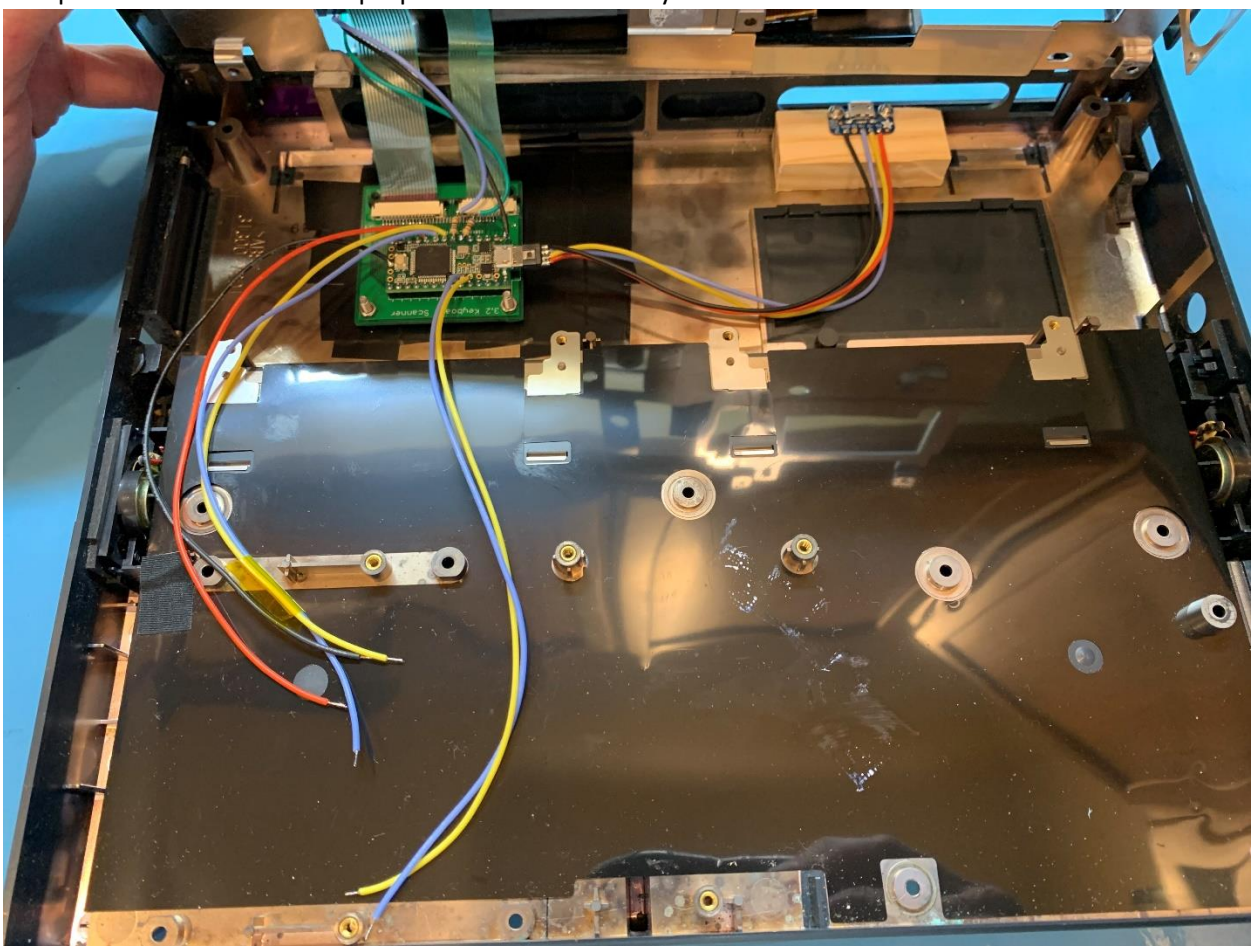
Project #1: Convert laptop base to a USB keyboard with a pointing device.

The LCD lid was removed at the hinge. The motherboard and other internal circuitry shown below was removed. The pointing device known as an “Ergo Trac” was retained.

There is a 16 pin and a 12 pin FPC cable from the keyboard. When looped back to the connectors underneath, the 12 pin has bottom contacts and the 16 pin has top contacts. The 12 pin only has 8 contacts needed for the keyboard column inputs as the other 4 pins are for keyboard language identification.



This picture shows the bare laptop base with the Teensy 3.2 connector board bolted down.



The base interior has conductive paint so electrical tape was placed under the Teensy connector board which is raised off the laptop base with plastic washers. Only two bolts are used to attach the board to the base as the other holes are blocked by the FPC cables. Foam double sided tape is underneath the connector section of the board. A Sparkfun microB USB breakout board was bolted on a block of wood so the external USB cable can be attached to the backside. Wires for USB +5 volts, Ground, D+, and D- were run to a micro B connector plugged into the Teensy. Other wires attached directly to the Teensy are for the CAPS and NUM Lock LEDs (with dropping resistors), the left and right mouse buttons, and the Ergo Trac. The keyboard cables are spaced so that the 16 and 12 pin FPC connectors will fit side by side on the 1mm pitch pads. The 12 pin connector is soldered to pads 1 thru 8. It is slid sideways so the 4 language ID connector pins are not connected to pads on the board. They are attached to the board with JB Weld epoxy to relieve the strain on the other 8 solder joints. The 16 pin FPC connector runs from pad 19 to pad 34. Pad 34 is connected to Teensy I/O #13 which also drives the LED on the Teensy. The LED was disabled by unsoldering its current limit resistor. This allowed the pin to be used as an I/O pin so the matrix decoder software would run. The results of the pin connection list was transferred to the matrix shown below. Surface mount pads 1 thru 34 on the board are listed first, followed by the Teensy I/O number used by the software.

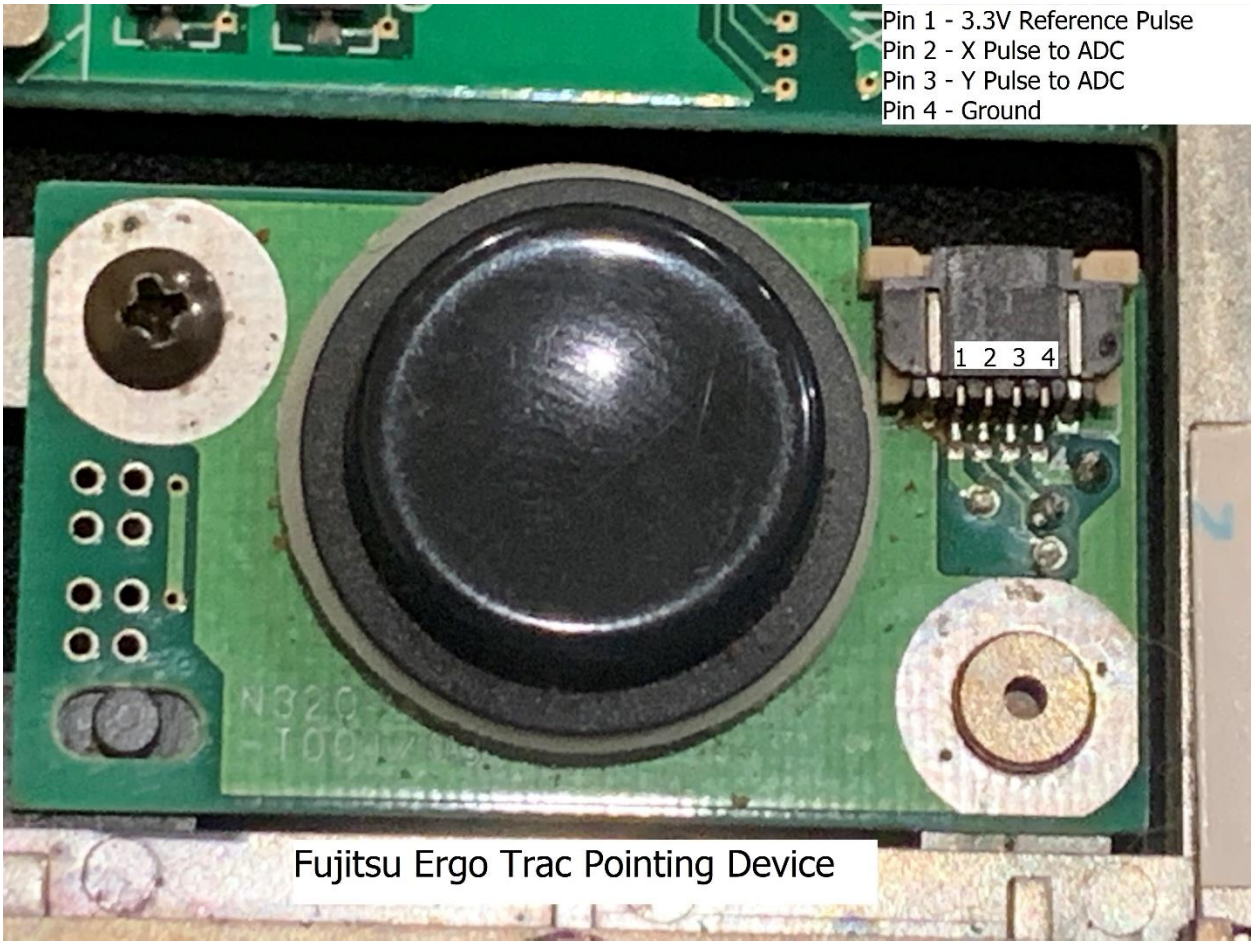
Fujitsu Keyboard Matrix

FPC#-IO#	1-23	2-0	3-22	4-1	5-21	6-2	7-20	8-3
19-33	Pg-dn	Pg-up	Prt-sc	Num-lk	pause	insert	home	bckspc
20-9	\	enter	F11	F12	delete	end		right
21-26	[	=	F10	0		minus	]	down
22-10	quote		F9	9	o	p	;	up
23-27	menu	/	8	i	k	L	period	left
24-11						gui		
25-28			7	u	j	m	comma	
26-12	b	c	F8	6	t	y	h	space
27-32	v	n	F7	5	r	f	g	
28-31	x	z	F6	4	e	1	d	alt-l
29-30	q	w	F5	3	2	s	a	
30-29							gui	
31-16				shift-r	shift-l			
32-15	cntrl-l							
33-14								fn
34-13	Cps-lck	tab	F4	F3	F2	esc	F1	Tilde

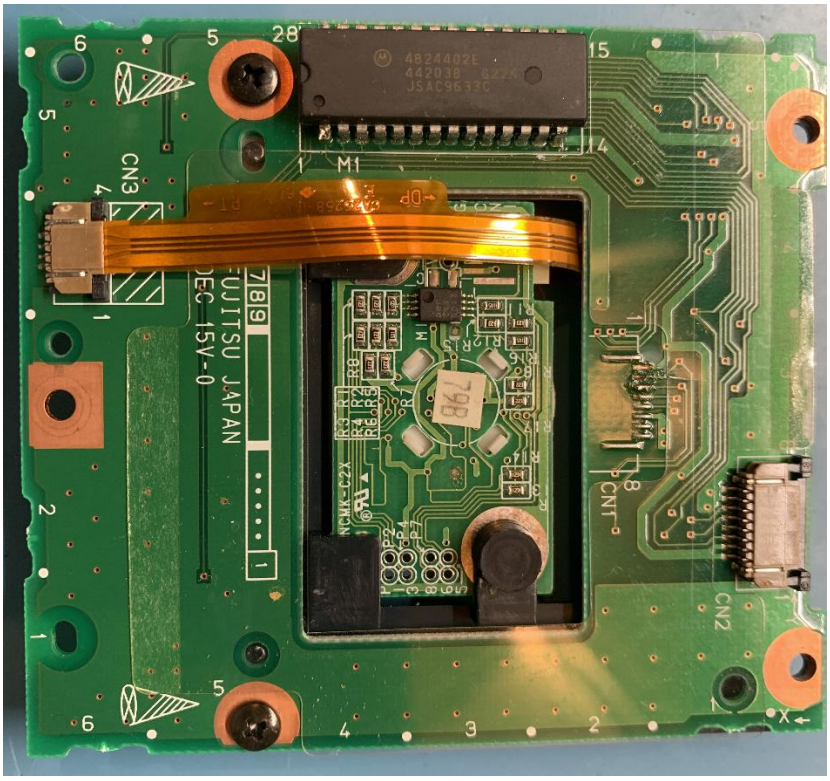
The keyboard has no number pad on the side but has an imbedded number pad centered on the “i” key.

When the NUM Lock key is pushed, the Teensy code uses a second matrix that has numbers for the key pad keys. If the shift key is also held down, these number pad keys become arrow keys. An LED will turn on to alert the user if the NUM Lock key has been selected. At power up, the software reads the NUM Lock LED status bit from the host and if turned on, a NUM Lock key press and release is performed. This turns off NUM Lock so the user starts out with a normal keyboard.

The Ergo Trac pointing device consists of 2 boards attached with a 4 pin FPC cable. The N320-?-T001/03 raw sensor (shown below) looks similar to the one described in the FID-828 data sheet with board number N310-?-X001/02. This board can be purchased on EBay and other sites.



The CA25290 controller board shown below looks similar to the CA20290 sold on New Egg and other sites. The 28 pin DIP on the controller board is marked Motorola 4824402E 442038 G22K.



I was unable to find a useful datasheet for this IC. I monitored what appeared to be the clock and data pins from the IC to the motherboard (see below) but was unable to cause any change in the data pattern when moving the sensor or pushing the mouse buttons. The motherboard was non-functional so the IC may not have been set up properly or did not receive the proper handshake signals. Because of this, I switched to directly reading the raw pointing device sensor.

Clock and Data signals on Motorola 4824402E pins

Scope measurements on the sensor board at the FPC connector showed that the controller applies a 500usec, 5 volt reference pulse on pin 1 every 80msec. The magnetic field detection circuit in the pointer translates the reference pulse to an X and Y pulse on pins 2 and 3, with ground on pin 4. At rest, the X output pulsed to 2.5 volts and the Y output pulsed to 2.9 volts. Applying pressure hard in each direction caused the pulse voltage to change per the following table:

Pin 2 Hard Left	1.9 volt pulse
Pin 2 Hard Right	3.1 volt pulse
Pin 3 Hard Up	4.0 volt pulse
Pin 3 Hard Down	1.8 volt pulse

With the raw sensor board disconnected from the controller board, pin 1 measures 288 ohms to ground so it only draws 11ma when driven with a 3.3 volt logic output from the Teensy. Pins 2 and 3 were wired to ADC inputs A10 and A11 on the Teensy. Teensy ground was wired to pin 4.

At startup, the Teensy code sends out a 3.3 volt logic pulse and reads the resulting X and Y pulse amplitude from the sensor. These values are saved as the “at rest” position. It is assumed that the user is not pushing on the sensor at power up. In the main loop, every 30msec, after the keyboard is scanned, the X and Y pulse values are read by the ADC. If either is more than or less than the “at rest” values (with an added noise zone), the cursor is moved using the Teensyduino USB mouse functions. The difference in the “at rest” value and the value currently read by the ADC determines the cursor speed. An additional speed control variable is used to slow the cursor down to normal speed. The parameters discussed above can be modified by holding down the Fn key and pushing the following function keys:

Fn-F5 = Toggle Ergo Trac On/Off. Wakes up turned On. Left/right mouse buttons always function

Fn-F6 = Change all Ergo Trac parameters back to the default values used at startup

Fn-F7 = Decrease cursor speed (takes several presses to notice a difference)

Fn-F8 = Increase cursor speed (takes several presses to notice a difference)

Fn-F9 = Decrease Noise Zone (too small and the cursor will move on its own due to noise)

Fn-F10 = Increase Noise Zone (too big and it will take excessive pressure to get any movement)

Fn-F11 = Recapture the Ergo Trac "at rest" X and Y values. Use this if temperature changes over time cause the cursor to start drifting.

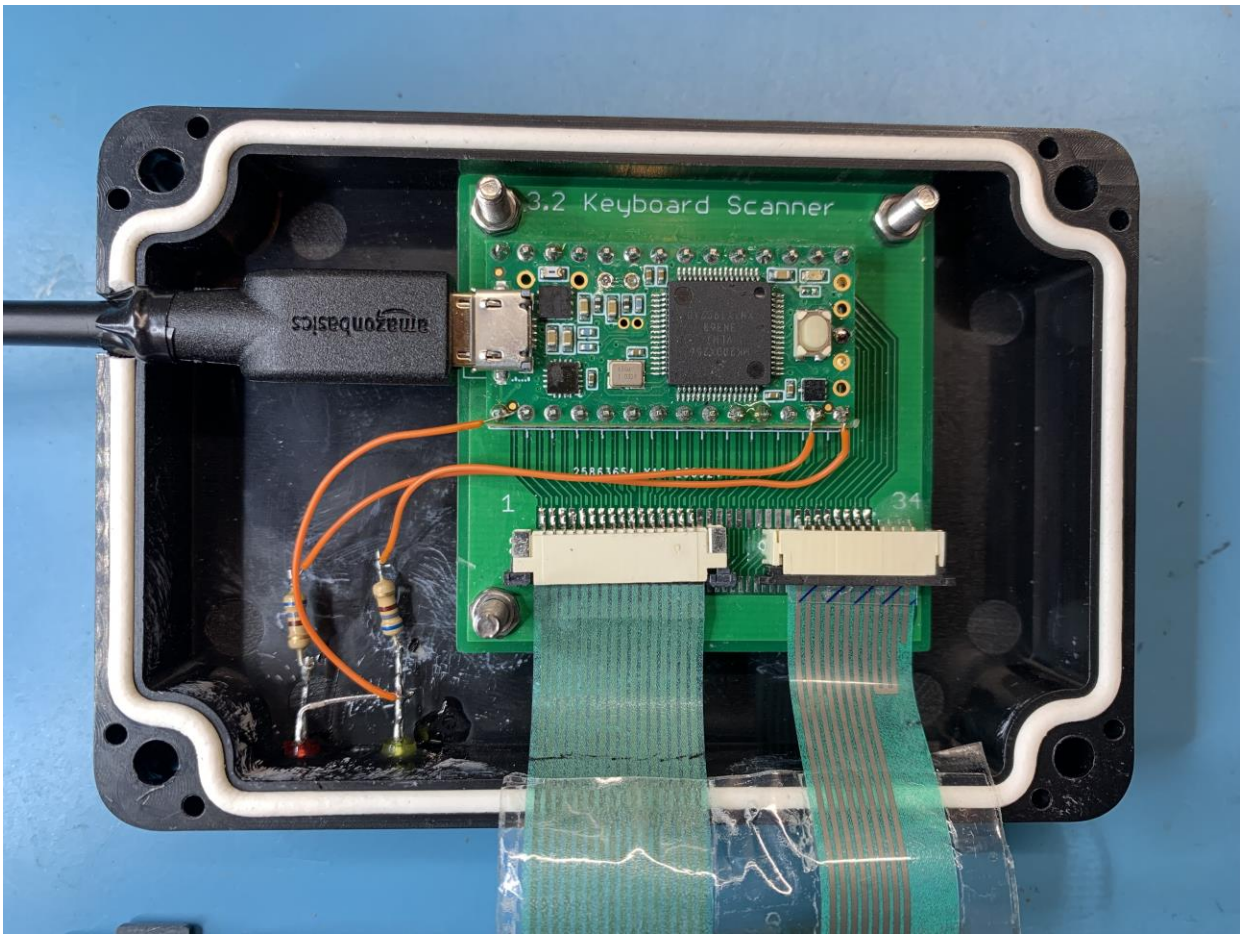
The Teensy 3.2 code for the USB keyboard with Ergo Trac is called “Fujitsu_KB_Ergotrac.ino” and can be downloaded from my repo.

Project #2: Convert a bare Fujitsu Lifebook 755Tx keyboard into a USB device using a Teensy mounted in a project box. This keyboard is fairly rigid and can sit directly on a desk or another laptop keyboard. A rubber pad should be cut to the size of the keyboard with 3 holes cut for the mounting posts on the bottom. The picture below shows the finished product.

The keyboard FPC cables do not loop back underneath so the orientation is opposite from the laptop base project. The 16 pin 1mm pitch connector has bottom contacts and the 12 pin has top contacts. The key matrix is flipped as well.

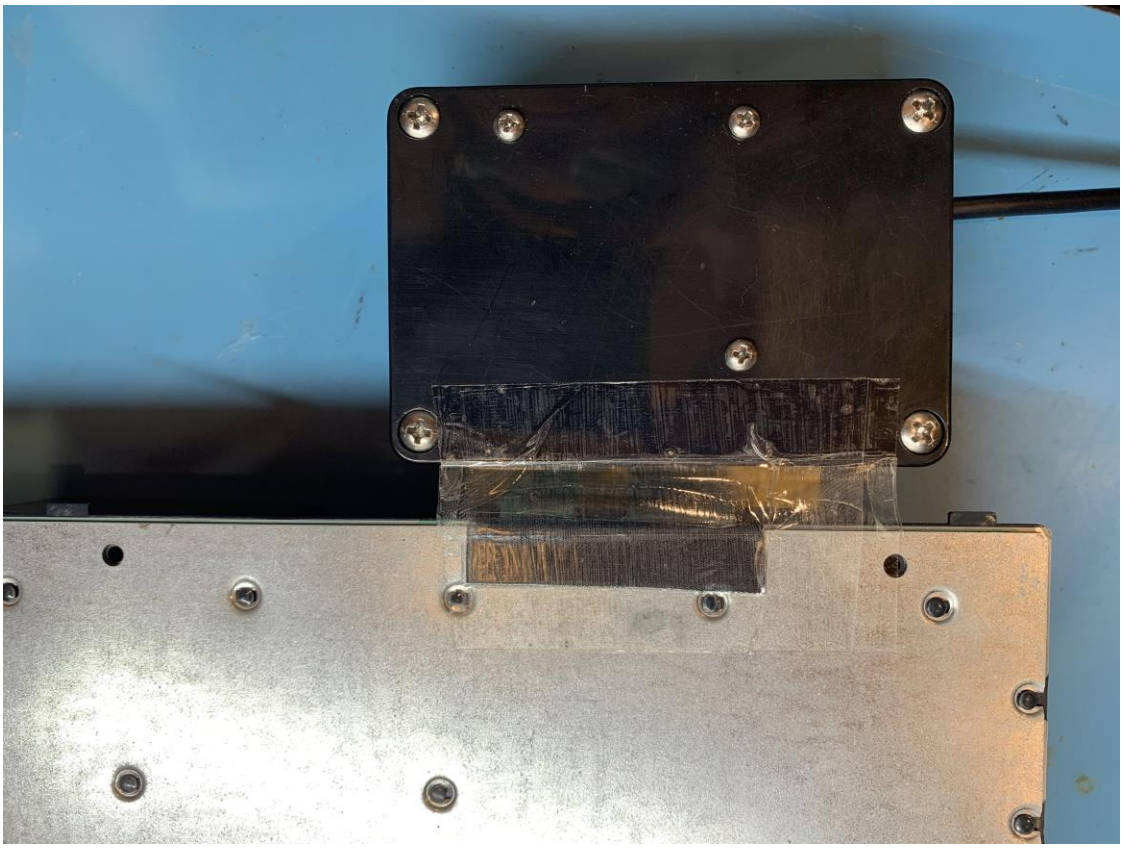
A close up of the box shows CAPS and NUM Lock LEDs next to the FPC cables that have been covered front and back with tape for protection.

The lids from two project boxes were used to reduce the height and to give a rubber gasket seal on each side to gently pinch the FPC cables. The following picture shows the interior Teensy 3.2 connector board and LED dropping resistors. The USB cable is a tight fit so it can't be accidently pulled out. The 12 pin FPC connector was slid to the right so that the 4 unused traces do not go to pads. These 4 pins were glued to the board with JB Weld epoxy. If this was not done, the connector would twist and possibly break when the slide drawer was pulled out. Because Teensy I/O #13 is used for the keyboard, the LED on the Teensy was disabled by removing its resistor.



A wire was run inside the trough so that the gasket was held halfway out (see below). The same was done on the top half of the case. The end result is a gasket pressing against the top and bottom of the FPC cables.

As a final means of protecting the FPC cable, tape was added from the bottom of the box to the keyboard in order to hold the two together.



Teensy 3.2 USB keyboard code “Fujitsu_Lifebook.ino” can be downloaded from my repo.