E-mu Ultra keyboard adaptor

An adapter to use USB keyboards with any E-mu Ultra sampler without the DWAM expansion.

This is, in essence, a USB to “almost PS/2” converter. The E-mu does not and will never understand USB, but we can take input from a USB keyboard and output what the E-mu is expecting to see from an attached AT or PS/2 keyboard.

The DWAM expansion contains an interesting circuit that turns a USART into a proper PS/2 port by multiplexing the RX, TX, and RTS lines together using a ’125 buffer. In PS/2 the host can request the keyboard’s attention and send data to it, but it doesn’t look like we need it except to respond to the initial probing. The probing is only used to decide whether or not to show the “ASCII keyboard connected” thing during boot, but if we can live with that we only need the clock and transmit pins. Whether or not a keyboard was detected during boot, the E-mu happily accepts keystrokes when sent.

Release notes, April 2023 (updated december 2023)

This project has been under way, in some sense, for (sigh) almost ten years. The first actually working, useful version was made around the end of last year. I am releasing instructions and firmware binaries now, lest it rot on my harddrive for another decade. There is no PCB at the moment. A few people have built these and reported success. I am releasing the source code under GPL in case that makes people more interested in taking things further. I have also learned that it might be possible to adapt this project for classic e6400s, see the note at the end.

Bugs

- Something’s weird at startup. I seem to need to go to a different page and back using the panel before the keyboard starts working.

Flashing the firmware

You should have the file `firmware.uf2`. Connect the Pico via USB **WHILE IT IS NOT AT ALL IN ANY WAY CONNECTED TO YOUR SAMPLER**, and flash as per the Pico instructions.

This has not been tested with the wireless version of the Pico. I don’t know if there are differences.

Hardware DIY

IMPORTANT: I do not take responsibility for any damage caused to your sampler, Pi, studio, house, relationships or self-esteem caused by the functioning or non-

functioning of the hardware described here. This is for people who understand what they're doing and can double-check a pinout, check for shorts before booting, and generally think for themselves. This section describes what I think I've done, but it might contain errors. This has worked with my sampler, but yours might be a different revision, or your star chart for this month might be wrong. If something goes wrong, I can offer my best attempt at consoling words but nothing further than that.

DO NOT UNDER ANY CIRCUMSTANCES PLUG THE PICO INTO A COMPUTER'S USB PORT WHILE VBUS IS CONNECTED TO THE SAMPLER, WHETHER THE SAMPLER IS POWERED ON OR NOT. If you do this the computer will power *some* of the sampler, but not all of it, and this can be Very, Very Bad. The software has been developed using a second Raspberry Pi Pico set up as a "pico-probe" to programme and listen for debug output so the Pi talking to the E-mu never saw USB power. My suggestion is to socket the Pico and remove it for firmware updates.

NB: While working with the prototype, I have seen the sampler crash when hotplugging some USB devices - screen blacking out and a handful of LEDs lighting up. I'm not exactly sure why it is, it could be that the device being plugged in causes a voltage drop, or it could be ESD. A reboot has fixed everything each time up until now, but for the time being, I would recommend against hotplugging even though the firmware supports it.

I've tested using level conversion on IO10 and IO11, but they're connected to a 74VHCT244 (U503) on the Ultra main board, and it should be fine with a 3.3V signal from the Pico. One person has built and adapter without level shifting and says it works.

The DWAM board hooks up to CN501. Pin 1 is marked on the board. All the odd-numbered pins (ie., the side of the connector that's closest to the front) are grounds. Please consult the relevant page of the Ultra schematics.

```
Pico GND  -> Ultra CN501 pin 21
Pico VBUS -> Ultra CN501 pin 22
Pico GP11 -> Ultra CN501 pin 24 (keyboard transmit)
Pico GP10 -> Ultra CN501 pin 26 (keyboard clock)
```

Probably the most stable way is to make a small perfboard or stripboard adaptor with a full 2x17 header and the Pico. For the USB connection, you can either run an extension cable/OTG cable directly from the micro USB plug on the Pico, or you can solder some wires to the test pads on the bottom of the Pico and connect that to a USB breakout board. My test setup used wires, here is a picture of the cable I made with the Pi pins to connect to listed.

Classic E-mu compatibility

Some pre-Ultra E-mus, such as the original e6400, were also upgradable with an ASCII keyboard port. I have not tried this at all and have no way to do so.

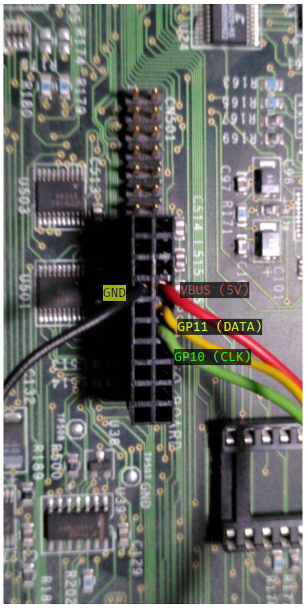

Figure 1: Image of hand-made cable to connect the E-mu Ultra to the Pi Pico, with color-coded cables

But the schematic that's floating around says the mainboard has a "J11" 14-pin header and in theory it might work.

```
Pico GND  -> Classic J11 pin 9
Pico VBUS -> Classic J11 pin 12
Pico GP11 -> Classic J11 pin 11 (keyboard transmit)
Pico GP10 -> Classic J11 pin 14 (keyboard clock)
```

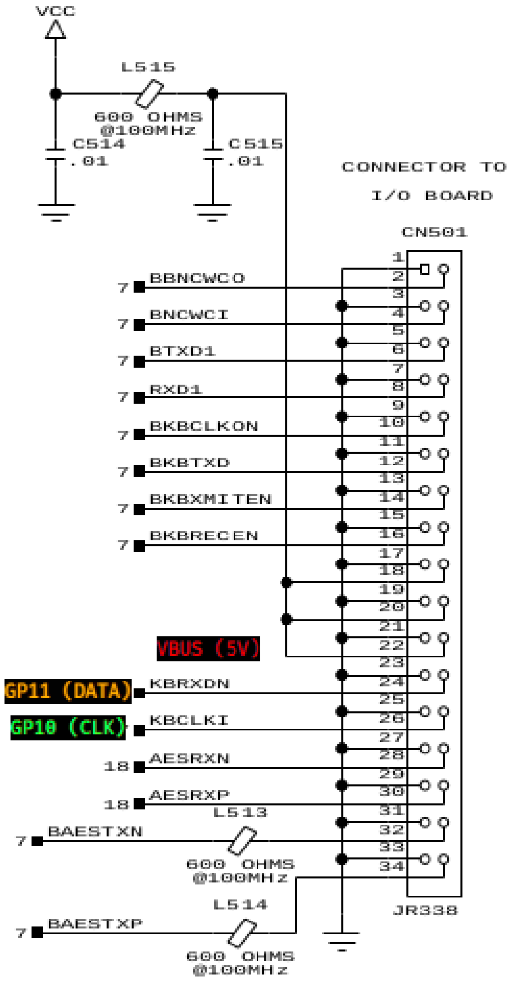

Figure 2: Excerpt from the E-mu Ultra schematic with the same color codes as above

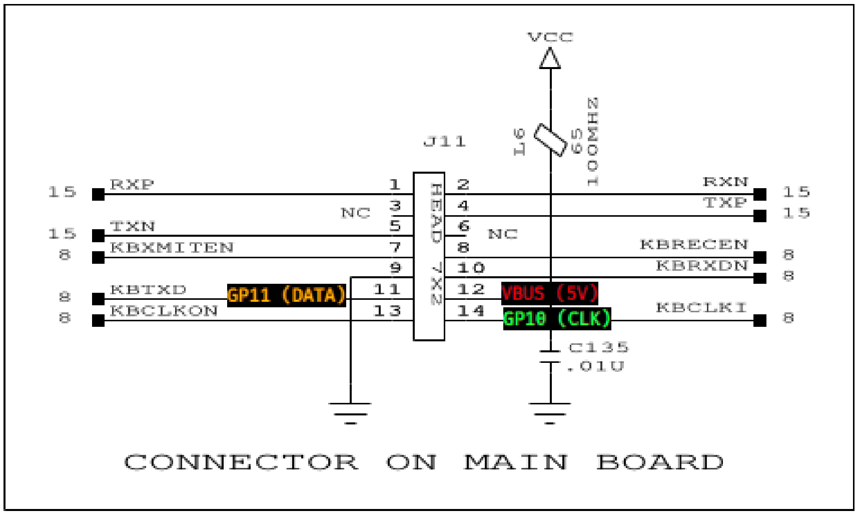

Figure 3: Excerpt from the E-mu Classic schematic