XT Keyboard Bluetooth Retrofit

Leading Edge DC-2014

Principle

The simplest way to integrate Bluetooth into any XT-era keyboard is to both convert from XT and transmit Bluetooth from the same circuit board. For that purpose, we're going to use the **Adafruit Feather 32u4 Bluefruit LE**, which retails for $30. The combination of a commonly used microcontroller and integrated latest-generation Bluetooth makes this the obvious choice.

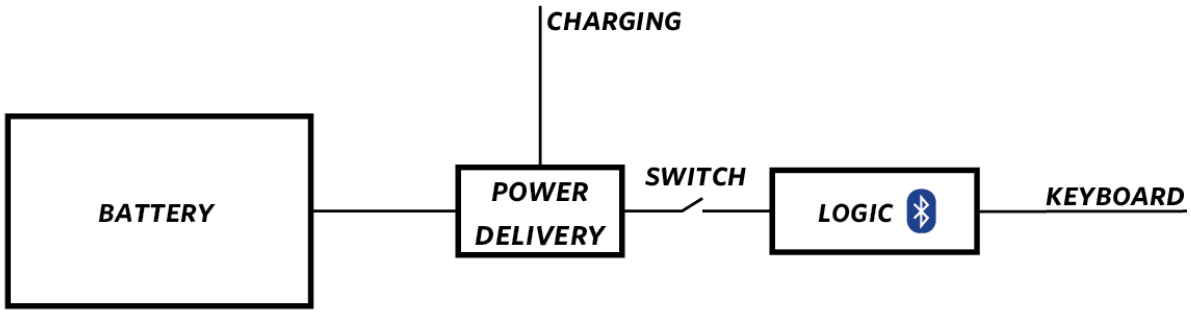
After the job of converting and transferring signals is complete, the only thing left to do is power the contraption. Depending on the size constraints (or freedoms!) inside your keyboard, you may have to adapt your layout, but the strategy should remain the same.

The very first order of business

Before you close up all the circuitry in your keyboard, you should flash the firmware on your Adafruit Feather. Scroll down to near the end of the document, and complete those simple steps first.

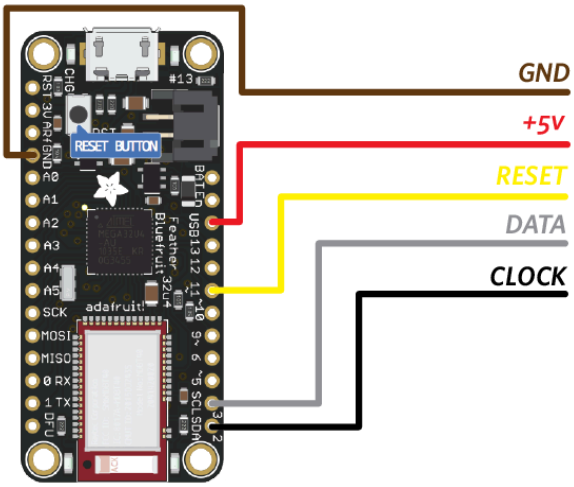
Circuitry Overview

The electronics can be broken down into three distinct components: battery, power delivery, and logic. They can be seen laid out in the simple diagram below.



Logic

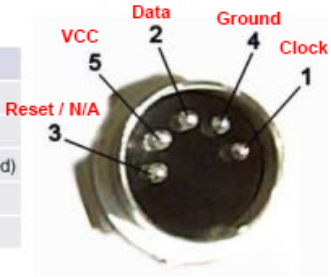
The logic is already sorted, it is the Adafruit Feather. It should be connected to the keyboard as necessary with power, ground, data, clock, and reset. If you've wired an XT adapter before, this will be a piece of cake for you, but if not, it is laid out here:



If you do not know which pin on your keyboard PCB corresponds to which line, then **do not guess!** Use a multimeter and compare the pinout of the pcb to the pinout of the XT connector at the end of the cable:

AT keyboard connector (DIN5)

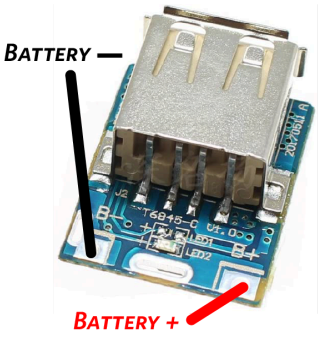
Connector Pin #	Purpose
Pin 1	KBDCLK (clock)
Pin 2	KBDAT (data)
Pin 3	KBRST (reset, not used)
Pin 4	GND
Pin 5	VCC (+5V)



Power Delivery

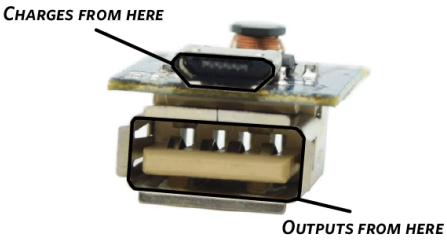
For our power delivery, we need to accomplish two goals. We need to provide a constant 5v supply for the Adafruit Feather, and we need to be able to charge the integrated batteries. Once again, such a circuit exists, and it's the exact kind used in commercial portable chargers. You can find it here: [HT4928S Lithium Battery Charger and Step Up Module]

In addition to a number of protection features, this board has USB built in already which makes it easy to connect to everything else inside the keyboard. There are three connections we need to make to the power delivery board.



First, we need to connect it to our battery. Solder the battery leads (described in more detail later) to the “B+” and B-” terminals on the power delivery board as pictured on the right.

Second, the power delivery board needs to output to the Adafruit Feather. This is also where the switch goes. If you have both male USB and male micro USB headers, you may choose to use those and wire a switch between the two. You can also cannibalize a micro USB cable and add a switch on the 5V line, which is what I did.



Third, the power delivery board needs to receive power to charge the keyboard. There are multiple ways to accomplish this. In my keyboard case, there was not enough room for the micro USB charging port so I fashioned a custom solution, and that is totally up to you.

Battery

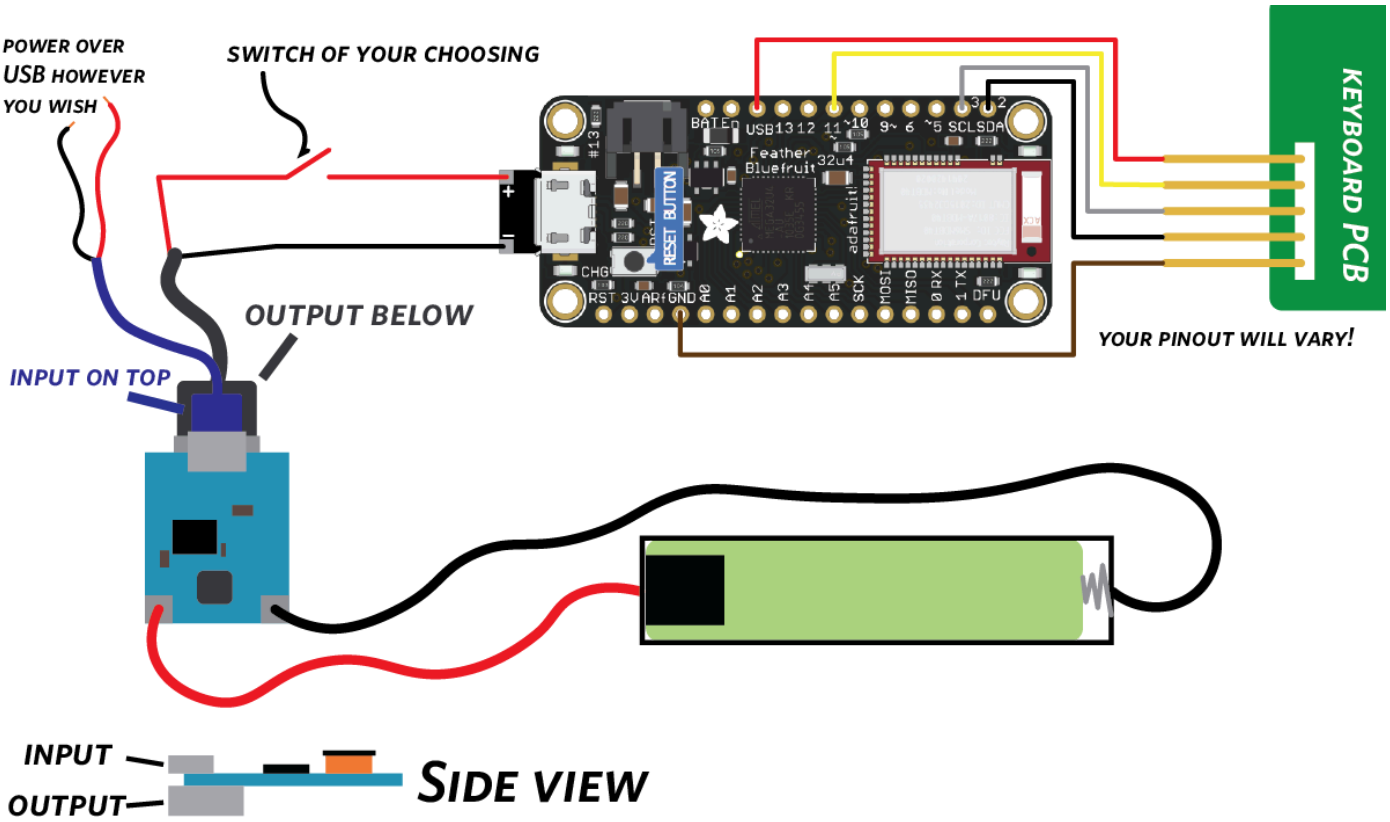
You have some flexibility when choosing a battery for your keyboard, but it must meet a small set of strict criteria:

1) The battery must be rechargeable
2) The battery must be a lithium-ion (Li-ion) or lithium-polymer (LiPo) battery
3) The battery must comfortably fit inside your keyboard case without being squished, bent, or crushed.

If your keyboard is thin yet spread out, you might opt for **a LiPo battery**, for example. If your keyboard has less of a thickness constraint, you might opt for one or more **18650 size batteries** instead. No matter which form factor you choose, make sure it has enough capacity, (multiple thousand mAh) and that it is from a reputable seller. **eBay and Amazon do not sell 18650 batteries. Only purchase 18650 batteries on dedicated websites, and only from real manufactures like LG, Samsung, Sanyo and Panasonic. No 18650 battery will have a capacity over ~3500mAh, and no 18650 battery will cost less than about $5.** Be smart, don’t get scammed and don’t get hurt.

If you choose a LiPo battery, it will already have leads attached that you can solder to the power delivery board. If you choose an 18650 battery, you will need to buy a package of holders like **this one**.

Full Circuit Diagram



Prebuilt Firmware (the easy way)

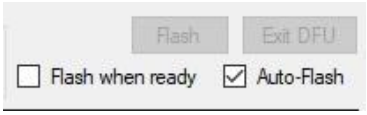
Should you like to go the easy route, you can download the Bluetooth firmware **here**. Right click the page and “save as.” You can view the layers in the layout here.

The steps to install it are as follows:

1. **Download** and install QMK toolbox
2. Open QMK toolbox and select the .hex firmware file that you downloaded

3. Set “MCU (AVR only)” to `atmega32u4`

4. Check “Auto-Flash”



5. Plug in your Adafruit Feather and press the reset button twice quickly. The firmware will be flashed automatically.

Programming your own Firmware

Modifying existing XT converter firmware to include bluetooth is simple with QMK, and the edits that need to be made are detailed here. It is assumed that you have a QMK environment set up and are capable of compiling the firmware yourself. If you don’t feel up for all that, the prebuilt firmware will work just fine.

The first edits that needs to be done happens in the rules.mk file

1. By default, the `BOOTLOADER` value will be set to `halfkay`, a commonly used bootloader, but the Adafruit Feather is based on another architecture, so the `BOOTLOADER` needs to be set to `caterina`
2. Also by default, the software will attempt to run the Adafruit Feather CPU at 16mHz, but this will not work, as the Adafruit Feather runs at 8mHz. As a remedy, the line `F_CPU = 16000000` should be replaced with `F_CPU = 8000000` and if no F_CPU line is present, it should be added.
3. In order for the firmware to fit on the Feather, it needs to be shrunk, so set `MOUSEKEY_ENABLE = no` to save space.
4. The next hurdle lies with a limitation of Bluetooth itself. NKRO is not possible, and as such it must be turned off. Set `NKRO_ENABLE = no` to solve the issue.

5. The final edit is to enable Bluetooth in QMK build. Add the line `BLUETOOTH = AdafruitBLE` to do so.

By now, the firmware is programmed to utilize Bluetooth, but by default it will output signals over USB whenever it is plugged in. This presents a problem since we are *powering* the Feather over USB.

To remedy this, an edit needs to be made to the keymap.c file. On the keymap, the key codes `OUT_BT` and `OUT_USB` need to be added to one key or another. It is suggested you relegate this function to a separate layer. **If your keyboard is not outputting over bluetooth, you likely need to set it back to** `OUT_BT`

Another cosmetic edit can be made in the config.h file. There is a line labelled `#define PRODUCT` which represents the Bluetooth name of the keyboard. You may choose to change it to whatever you wish.

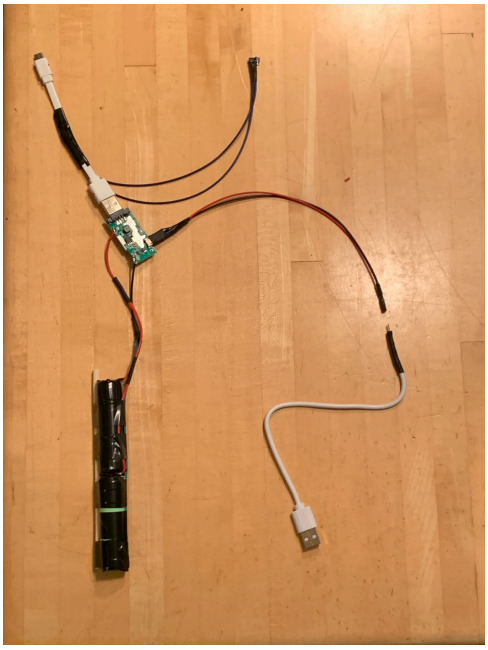
After you are done, rules.mk should look like this and you are ready to compile the firmware and flash it with the same method described in the prebuilt firmware section.

```
MCU = atmega32u4

# Bootloader selection
#   Teensy       halfkay
#   Pro Micro    caterina
#   Atmel DFU    atmel-dfu
#   LUFA DFU     lufa-dfu
#   QMK DFU      qmk-dfu
#   ATmega32A    bootloadHID
#   ATmega328P   USBasp
BOOTLOADER = caterina
F_CPU = 8000000

# Build Options
#   comment out to disable the options.
#
BOOTMAGIC_ENABLE = no   # Virtual DIP switch configuration
MOUSEKEY_ENABLE  = no  # Mouse keys
EXTRAKEY_ENABLE  = yes  # Audio control and System control
CONSOLE_ENABLE   = yes  # Console for debug
COMMAND_ENABLE   = yes  # Commands for debug and configuration
NKRO_ENABLE      = no  # USB Nkey Rollover
XT_ENABLE        = yes
CUSTOM_MATRIX    = yes
BLUETOOTH = AdafruitBLE
SRC += matrix.c

# Optimize size but this may cause error "relocation truncated to fit"
#EXTRALDFLAGS = -Wl,--relax
```

Photos

Left: example of a custom charging port and power switch that fits where the original cable used to be.

Below: The keyboard pre-conversion



Left: Battery, charge circuit, switched USB cable and custom charging cable. Note: a visually different but functionally identical charge circuit was used.

Right: All electronics inside the keyboard