

Additional Clock Output Mod for Banana Split

Although the Banana Split is a zero-latency MIDI thru box without any signal processing, it does contain a microprocessor – this processor simply “listens in” to the MIDI traffic and blinks the blue (activity) and yellow (clock) LEDs.

The processor has a couple of spare output pins that are made available as pads on the Banana Split PCB and from firmware version 3 (released in June 2018) these pads can be assigned various pulse clock functions.

This note only applies if you have version 3 firmware.

How do I know if I have version 3 firmware?

If you purchased the device after Jun 2018 you probably do, however there is no external marking to indicate this – what you need to do is send a special sysex file to the Banana Split and see if it responds.

You can download the sysex files using this link: <https://github.com/hotchk155/banana-split/tree/master/sysex>

The following files are available

sig_8th.syx	One clock pulse per 8 th note (Korg volca default)
sig_16th.syx	One clock pulse per 16 th note (most Eurorack sequencers)
sig_24ppqn.syx	One clock pulse per 96 th note (DIN SYNC-24)

Use any one of the files to perform the check. Make sure you click through to the individual files web page on Github and click on the Download button (do not right click link / save as or you'll save the web page instead of the sysex file!)

When you send the sysex to the Bananasplit, the yellow and blue leds should blink alternately until you power down the Split. If nothing happens (except blue light coming on briefly) then you must have an older Split pre version 3 of the firmware. Or possibly you have an issue sending sysex – check your settings, especially if the blue light did not come on.

What can I do if I don't have Version 3?

Unfortunately you will need to program the new firmware on to the chip using a special programmer interface. The chip is a PIC12F1822 and can be programmed using a PICKit2 programmer or equivalent. You will also need to solder wires temporarily to the chip for the programming, so this is probably not a job for beginners. If you need assistance let me know, I can update the firmware for you if you cover postage costs.

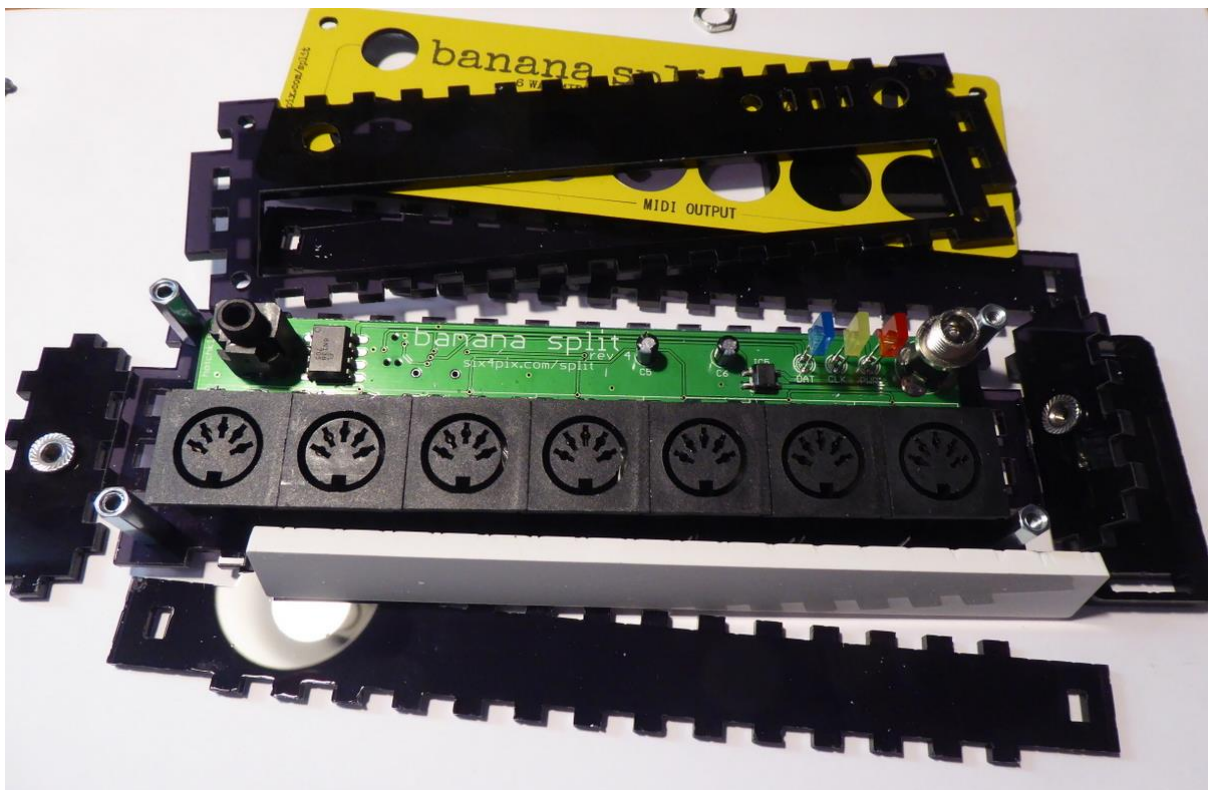
The firmware HEX files can be found here <https://github.com/hotchk155/banana-split/tree/master/firmware/Release>. Once again click through and use the Download button to get the raw file.

OK, I have the right firmware. How do I get at the clock signals?

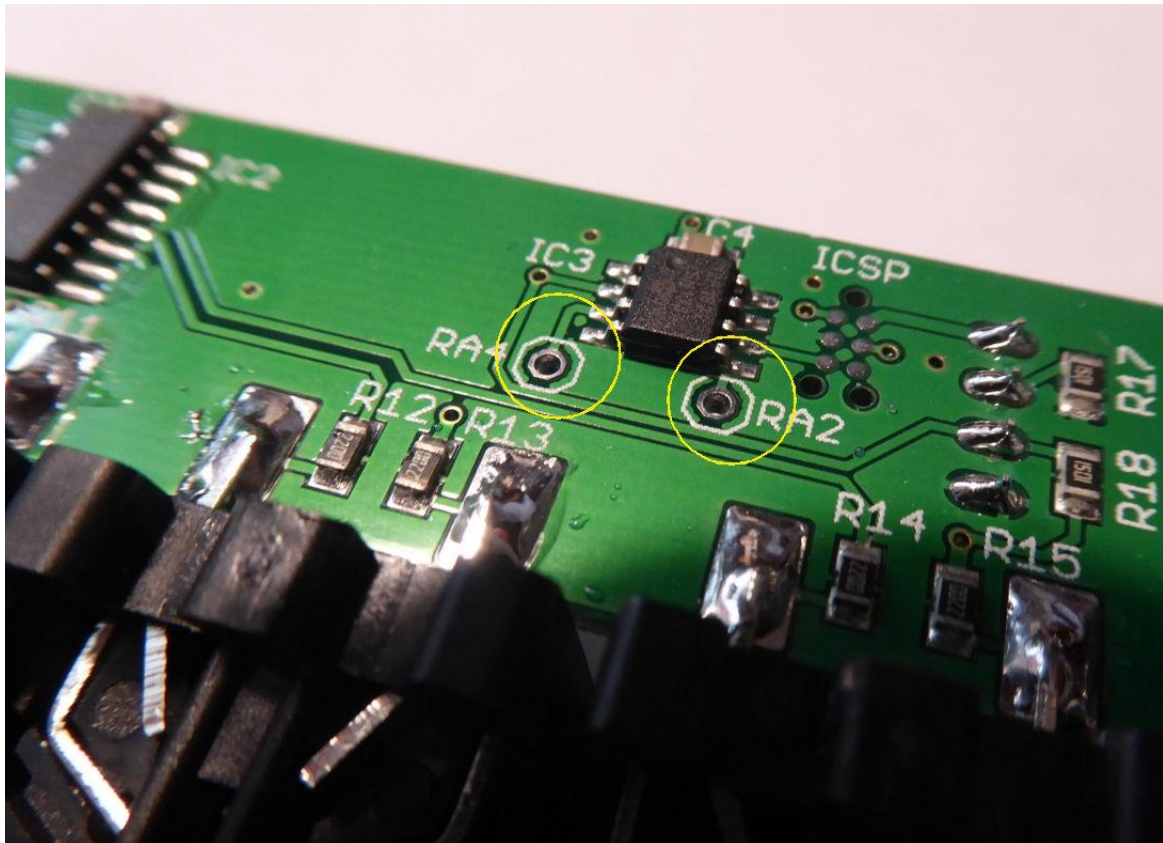
You need to disassemble the Split. Using a 2.5mm hex key, remove the bolts from the face plate and the ends. Try not to push down on the end bolts when turning them, since you could push out the flanged nuts that they screw into.



Remove the face plate and the nut from the power socket and gently pull everything apart to extract the PCB.

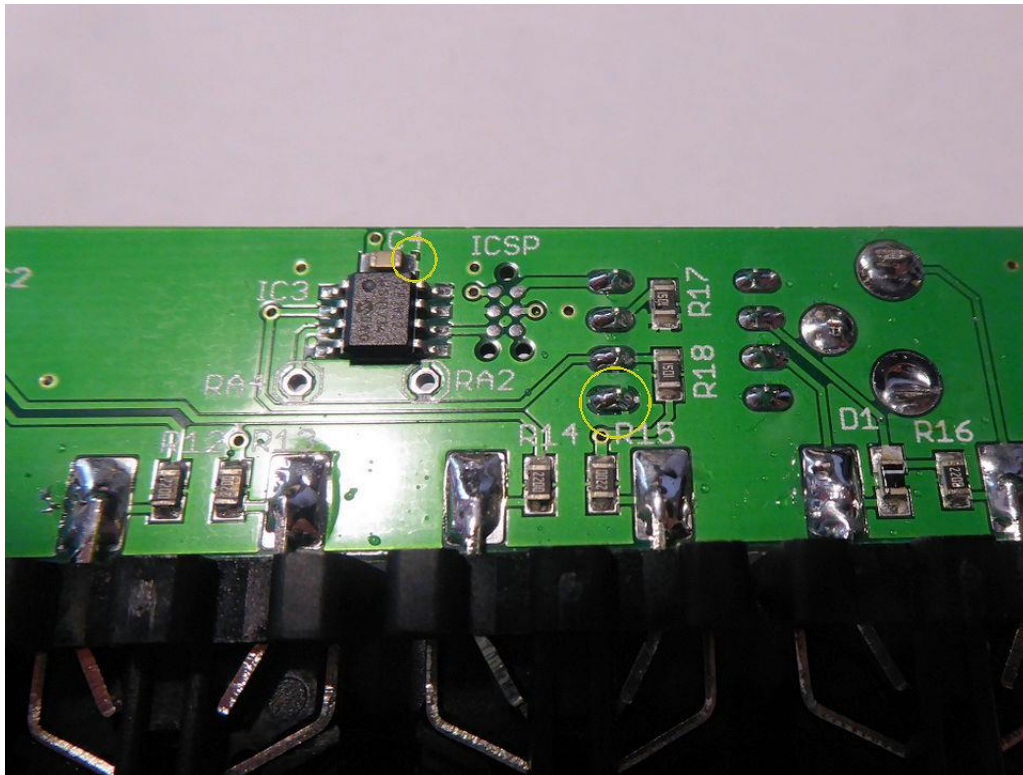


Turn the PCB over and look for the pads marked RA4 and RA2 next to the smaller IC.



- Pad RA4 is a 5V CLOCK signal, with a clock rate selected based on the last sysex file that was sent
- Pad RA2 is a 5V RUN signal, which is HIGH when a MIDI clock START or CONTINUE message is received until a STOP message is received.

To use these signals, connect wires to the pads. You will also need a ground connection. See the photos below for the closest grounding points: The best is probably the ground pin of the through hole opto isolator IC (next to R15) but you can also use the ground side of cap C4 or the centre pin of any of the MIDI output sockets (on the other side of the board)



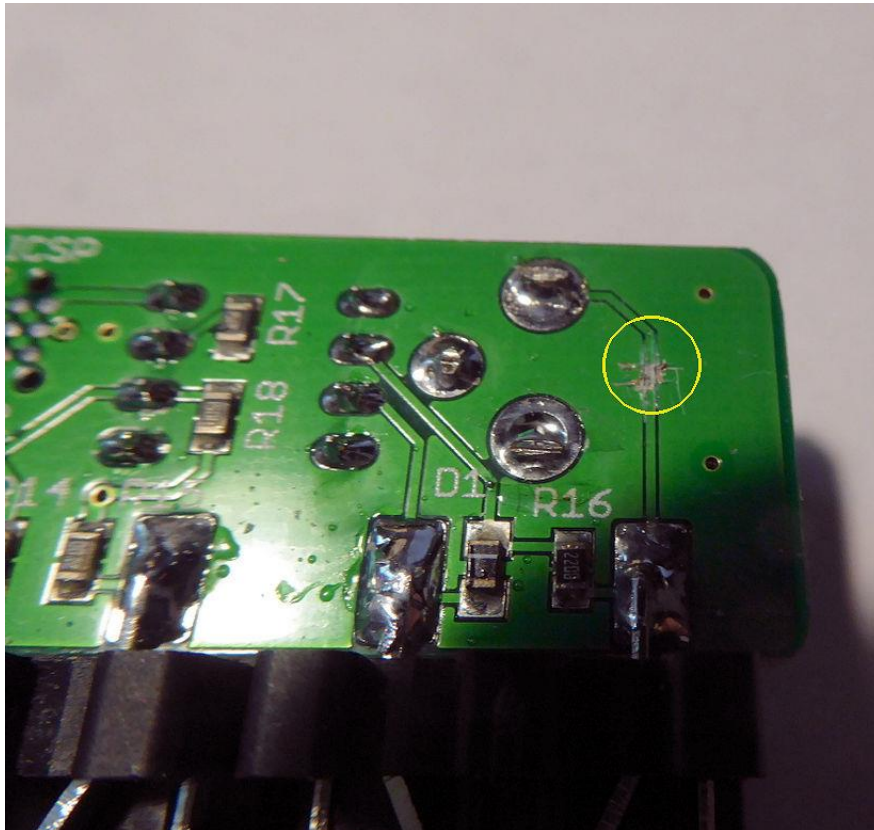
I advise that a current limiting resistor of around 1k ohm is placed in series with each signal that you break out to the outside of the case. Otherwise damage to the Splitter could be caused by short circuits (such as during plugging in a jack) or accidental connection to an output socket on another device.

If I am not using the 3.5mm MIDI jack, can I use it as a clock out?

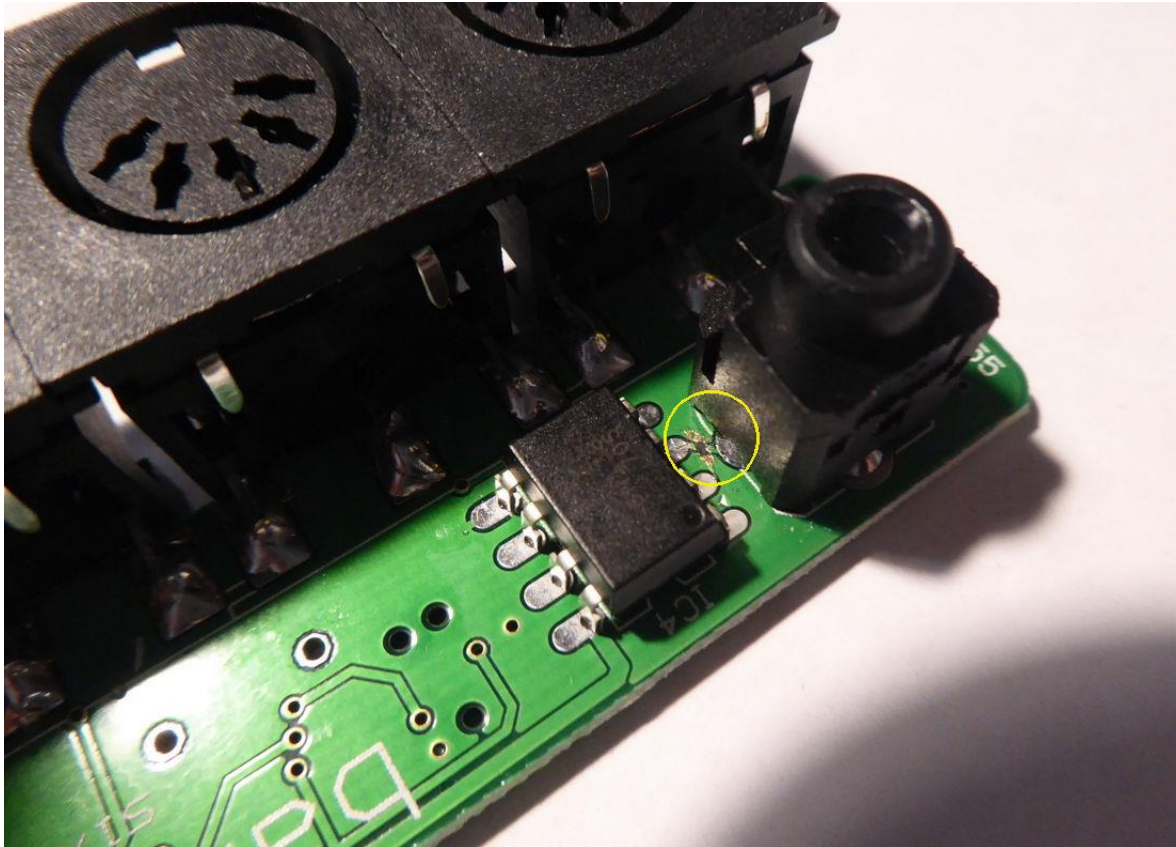
Yes! why the hell not? However be warned this is a one-way trip – **you will only be able to use the 5-pin MIDI input after making this change.**

First of all you'll need to cut the two PCB tracks that are connected to the socket. Use the tip of a scalpel to make a couple of cuts across the track then scrape out the copper between and clean up the area to remove any copper debris. Use a multimeter to make sure that there is no short between either end of the cut track and the ground plane.

The first cut is on the back of the board, to the track that connects with the top pin of the socket.

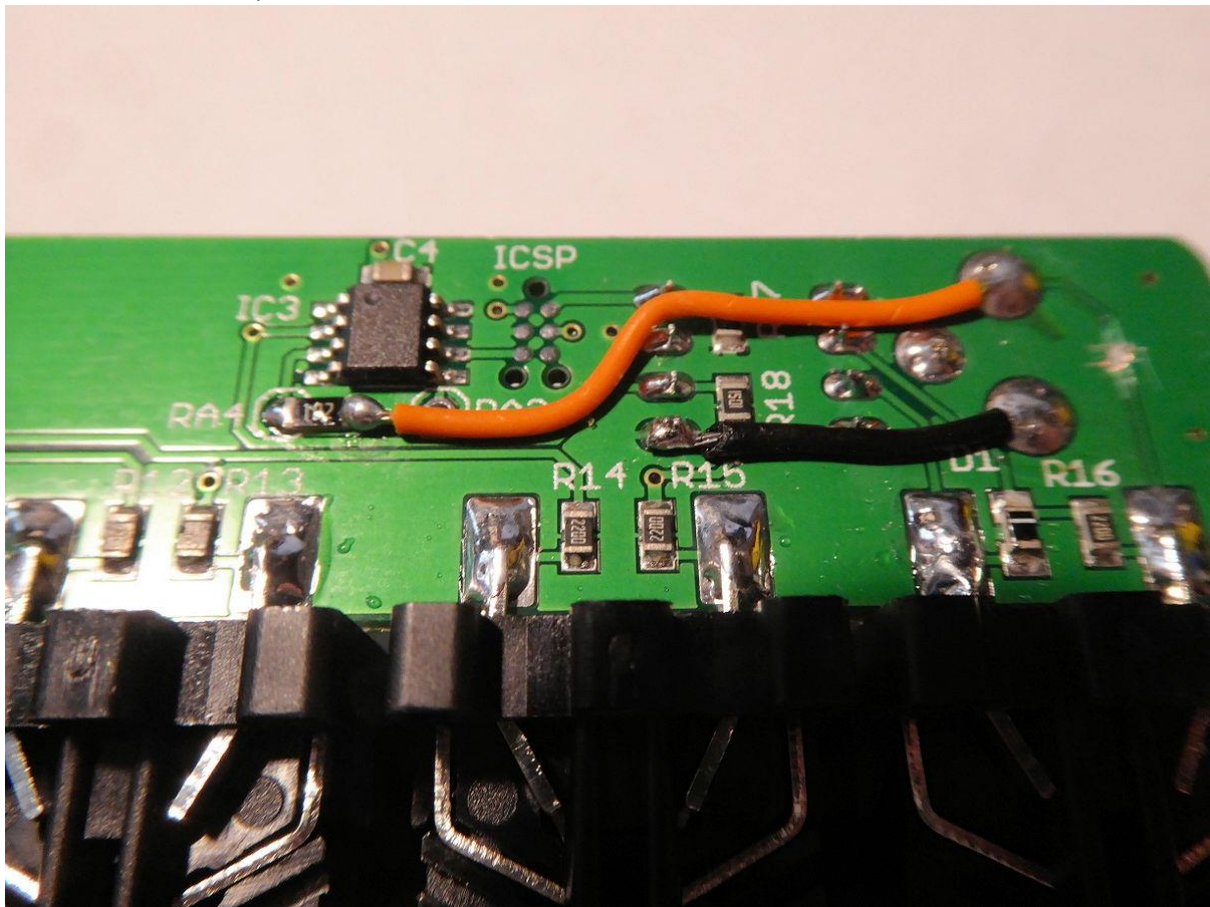


Second cut is between the socket and opto-isolator IC on the top of the board

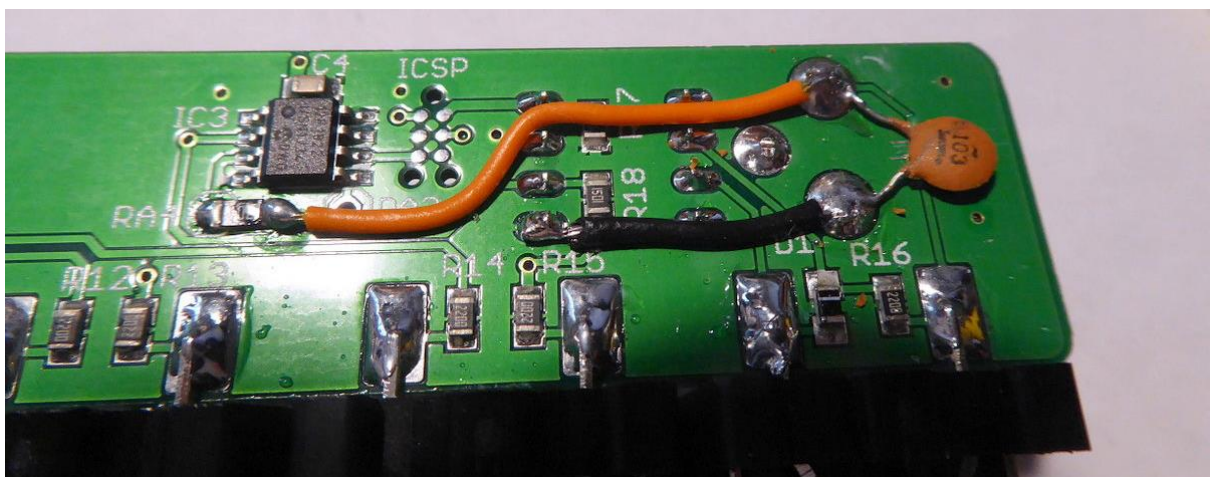


Solder a 1K resistor to the RA4 pad. A sub-miniature (0.125W) through-hole resistor or a 1206/0805 surface mount resistor should work just fine. Here I used a 1206 resistor.

Wire the other side of the resistor to the top pin of the socket and the ground pin of the opto-isolator to the lower pin of the socket as shown here



I have found that the Korg Volcas can sometimes “glitch” on a hard edged clock and register multiple ticks to a single pulse. This can be fixed by smoothing with a low-pass filter arrangement by adding a cap after the 1k resistor. I used a 10nF ceramic capacitor across the output terminals.



Now reassemble.. when screwing end panels be careful not to push the flanged nuts out of their holes. **Do not overtighten the bolts which fix the front panel – you will cause it to crack.** It is best to carefully tighten these bolts until they just start to tighten (i.e. a little over finger tight)

