



*JBardwell*

# JBardwell F4 AIO

*Betaflight Flight Controller*

**Pilot's Manual**

**V2.1**

**For Hardware Revision 2.0**

# RaceDayQuads / JBardwell F4 AIO

---

I'm Joshua Bardwell and this is the flight controller I always wished somebody would make. It's got all the major features that Betaflight supports. A clean and clear layout. It's backed by RaceDayQuads' top-notch customer support. It's affordably priced. Oh, and let's not forget, technical support from none other than me: the FPV Know-It-All himself.

Input Voltage: .....	3S to 6S	Manual written, and feature set selected by Joshua Bardwell.
Processor: .....	STM32F4	Superb board layout allowing for easy and clean builds.
Gyro: .....	MPU6000	9v filtered output power for vTXs. Lower output voltage increases the filtering of the regulator for the cleanest video. Vbat output is also available.
Power Outputs:.....	2x 5v 1x 3.3v 1x vBat 2x 9v filtered	All connections via solder pads for secure connection and high durability.
Regulator Rating:.....	1.2 amps each (5v & 9v)	Dedicated plug for easy connection to 4-in-1 ESCs.
Form Factor:.....	30.5mm square	Dedicated FPV Camera Control pad. Just solder it up. No fussing with resistors and capacitors!
Current Sensor: .....	240 Amps	On-board inverter for SBUS and SmartPort protocols. No "uninvert hack" or resource reassignment required.
Betaflight Target: .....	CLRACINGF4	
UARTs:.....	4 (2 inverted)	
Features:.....	Programmable LEDs FPV Camera Control SmartAudio Buzzer Output	

With the v2.0 board, we've made a few improvements. Most importantly, the 9v regulator has been redesigned to make it as robust and reliable as possible. The defect that plagued previous versions of this board has been eliminated. The current sensor max has been increased to 240 amps after reports that some users were exceeding the 130 amp maximum on previous boards (YOU BEASTS). The current sensor has also been factory calibrated so that you don't have to go through a tedious manual calibration process to get your amps and mAh accurate. Finally, the 4-in-1 plug has been upgraded to support both analog current sense and ESC telemetry, for compatibility with all ESCs.

# Functional Quick-Reference

---

The table below shows the UART and solder pad assignments for major features. If multiple UARTs are shown, choose one, depending on your exact feature set. UARTs are listed with the suggested “best” choice top-most and alternate choices below.

Keep mind these restrictions:

**A single UART cannot be used for more than one function.** For example, if you are using an iBus receiver on RX6, then TX6 cannot also be used for ESC telemetry.

**You cannot use both TX and RX on the same UART for two different functions.** For example, you cannot use ESC Telemetry on RX4 and SmartAudio on TX4 at the same time.

Function	UART	Pad
SBUS Receiver	1	S
iBus, SpekSat, or other non-inverted receiver protocol <i>SpekSat receiver bind mode only supported on RX6</i>	1 6 4	S R6 RX4
Crossfire Receiver	6 4	T6/R6 TX4/RX4
SmartPort Telemetry	3	T
iBus Telemetry, Spektrum Telemetry, or other non-inverted telemetry	6 4	R6 RX4
vTX Remote Control (SmartAudio, Tramp Telemetry)	4 6	TX4 T6
Programmable LEDs	N/A	LED
ESC Telemetry <i>RX4 pad at corners of the board is preferred for individual ESCs.</i>	4 6 1	RX4 R6 SBUS

The board has been designed so that most pilots will be able to use all the functions they want, right out of the box. SoftSerial is available for less common combinations, or if you accidentally damage one of the pads on the FC. Enabling SoftSerial is described later in the manual.

# Board Layout Quick-Reference

---

The table below shows the intended purpose of each of the pads on the board. Detailed instructions for wiring up the board are later in the manual.

## TOP OF BOARD

ESC-	Battery voltage negative, wired to ESC.
ESC+	Battery voltage positive, wired to ESC.
G	Ground
S1, S2, S3, S4	Motor signal outputs 1-4.
S	Serial receiver (SBUS, iBus, SpekSat) input. UART1. Can be used with both inverted and non-inverted protocols, but SpekSat binding don't work on this pad. Can also be used for ESC Telemetry if you're freaky like that.
T	SmartPort telemetry input (inverted) UART3.
3.3/5	5v or 3.3v output for receiver. Use solder bridge on underside of board to select voltage. This pad will receive power when the FC is plugged into USB, to allow you to work with your transmitter without plugging in the LiPo.
9V	9v filtered output for FPV camera and vTX. Max 1.2 amps.
R6/T6	UART6 transmit and receive pad. Use this for non-inverted receiver input such as iBus or SpekSat. iBus telemetry and SpekSat binding work on this pad. Can also be used for other receive functions like ESC telemetry. This is also the preferred pad for Crossfire receiver.
CAM_S	Video signal input from FPV camera.
CAM_C	FPV Camera OSD Menu Remote Control function. Connect to camera OSD input pin.
LED	Programmable LED output pin. Usable with Betaflight LEDStrip feature. Also, can be reassigned to SoftSerial via Betaflight resource remapping feature to provide one additional UART. <i>Instructions for this are later in the manual if you need them.</i>
TX4 / RX4	UART4 TX and RX pads. TX4 is primarily intended for use with vTX remote control function such as SmartAudio or Tramp Telemetry. iBus telemetry will not work on this pad.
BUZZ-	Negative leg of piezo buzzer.
BUZZ+	Positive leg of piezo buzzer. This pad can also be used as a generic 5v output if need be.
VCC	Battery voltage. Primarily intended to be used for powering a video transmitter, since some vTX do better on vBat than on 9v output. Can also be used to power LEDs if they are rated for battery voltage.
VIDEO	Video out to FPV video transmitter.

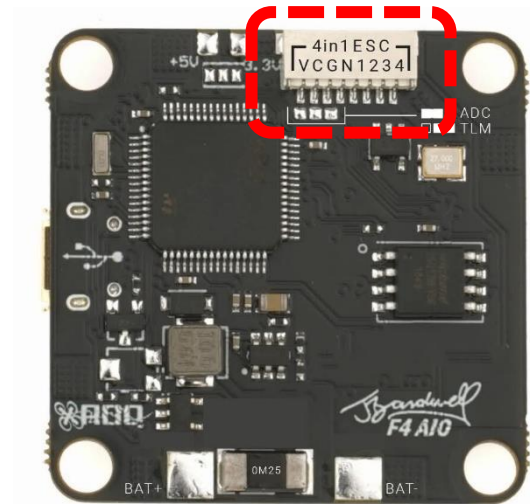
## BOTTOM OF BOARD

5v	5v output for accessory devices. Max 1.2 amps total across all 5v pads.
BAT+	Main battery lead positive wire. <i>If using a 4-in-1 ESC, connect battery lead to the ESC, not the JBF4.</i>
BAT-	Main battery lead negative wire. <i>If using a 4-in-1 ESC, connect battery lead to the ESC, not the JBF4.</i>
+5v/3.3v	Solder bridge for selecting 5v or 3.3v output to receiver pad on top of board.
ADC/TLM	Solder bridge for selecting analog current or ESC Telemetry for 4-in-1 plug

## 4-IN-1 PLUG SOCKET

V	Battery Voltage.
C	This pin has one of two functions, depending on the state of the ADC/TLM solder bridge.  When ADC/TLM solder bridge is set to ADC, this pin is used for analog current sense input from external current sensor installed on the 4-in-1 ESC. When used, this input replaces the internal current sensor on the JBF4 main battery lead pads.  When ADC/TLM solder bridge is set to TLM, this pin is connected to UART 4 RX and is used for ESC Telemetry function.
G	Ground.
N	No function. Remove this wire from the plug.
1, 2, 3, 4	Signal wires for motors 1, 2, 3, and 4.

The 4-in-1 plug makes connecting the JBF4 to a 4-in-1 ESC easy. The plug contains connections for power, ground, motor signal 1-4, and analog current sensing or ESC telemetry.

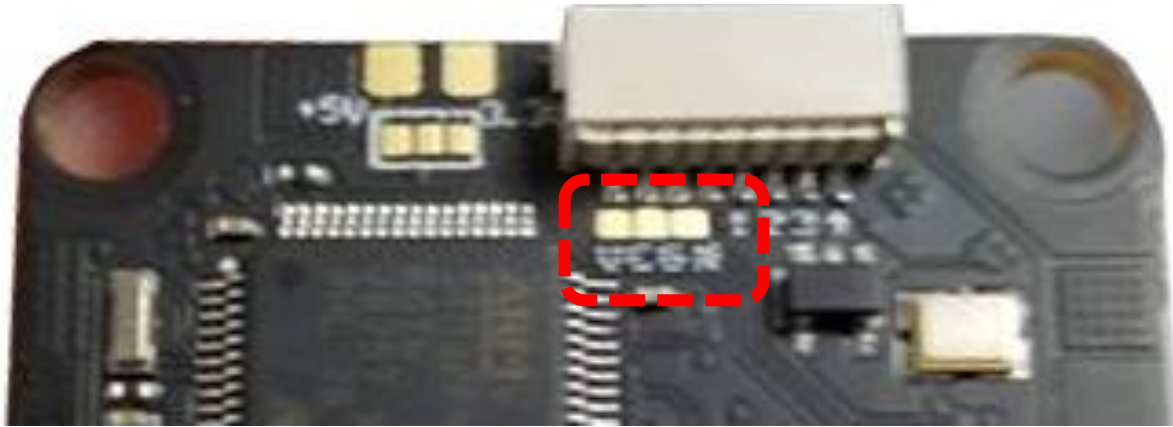


Since there is no universal standard for 4-in-1 ESC plugs, you cannot simply plug your ESC into the JBF4 and assume that everything will work. In fact, you might fry your ESC and/or the JBF4 if you did this! The plug on the JBF4 has been designed to be directly compatible with recent Airbot ESCs, however even if you are using one of these ESCs, **you must manually verify that the pinout is compatible before you plug in the ESC**. Damaging the JBF4 or your ESC by plugging in incompatible ones will not be covered under warranty.

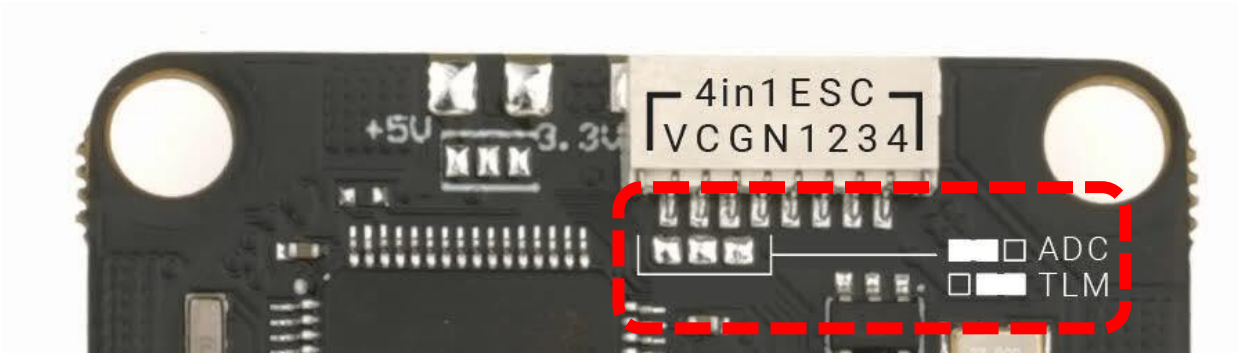
If you need to make your own custom wire harness to connect your ESC to the JBF4, RaceDayQuads sells a pre-crimped wire header set that is perfect. It can be purchased here: <https://www.racedayquads.com/products/sh-1-0-silicone-cable-set>

**Should you use analog current sensing or ESC telemetry?** If your ESC only supports one of the two, then that's the one you should use! If your ESC supports both, then we suggest using analog current sensing because it frees up UART4 for a different function. ESC Telemetry reports additional stats including motor RPM, but Betaflight doesn't really use those stats for anything, so there isn't much advantage to it.

The first production run of v2.0 boards had a silk screen as shown below. We decided that this silk screen might confuse some people. The silk screen is labeling the pins in the 4-in-1 ESC. The three solder pads are not V, C, and G. The solder pads are the solder-bridge for selecting analog vs. ESC telemetry for the C pin.



Below is what later runs of the board look like. This will be shown throughout the rest of the manual.



# Pre-Build Worksheet

---

This worksheet will help you document which UARTs and pads are going to be used for which functions on your build. Figuring this stuff out before you pick up a soldering iron will save you time and prevent mistakes.

The table below shows possible choices of UART and physical solder pad for each function. Recommended choices are highlighted in bold, however not all combinations are valid. For example, if you use RX4 for ESC Telemetry, you cannot also use TX4 for SmartAudio because each UART can only be used for one function.

Use the worksheet below to figure out which UARTs you will use for which functions. Circle the UART#/Pad Name for each function. Cross out lines for functions you will not be using.

Resource	UART # / PAD NAME		
SBUS Receiver	<b>1/S</b>		
Spektrum Receiver	1/S	4/RX4	<b>6/R6</b>
Crossfire Receiver		4/TX4+RX4	<b>6/T6+R6</b>
SmartPort Telemetry		<b>3/T</b>	
iBus Telemetry			<b>6/R6</b>
Spektrum Telemetry	<b>1/S</b>	4/R4	6/R6
vTX Remote Control		<b>4/TX4</b>	6/T6
ESC Telemetry	1/S	<b>4/RX4</b>	6/R6

# Soldering Tips

---

There's an uncomfortable truth in this hobby: a whole lot of us are much better with a transmitter than we are with a soldering iron. We have designed the JBF4 AIO to be resilient, but any board can be damaged if it is soldered on incorrectly. Before you take a soldering iron to your board, here are some tips to help make sure you're not making a call to support, trying to convince them you, "have no idea how that pad lifted off I barely touched it with the iron and it just came off!"

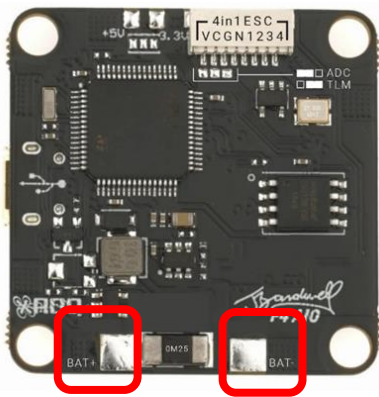
1. We suggest using a temperature controlled iron if at all possible. This type of iron has a temperature sensor in its tip and adjusts its output power to keep the tip at a constant temperature. If your iron doesn't have adjustable temperature, you will need much more skill and experience to get good results. You don't have to spend \$100 on a high-end Hakko or Weller iron to get temperature control. There are budget irons available in the \$30 price range that will do the job.
2. We suggest a temperature of about 400°C / 750°F. Some beginners believe that a colder temperature is safer because it's less likely to damage the board, but this isn't true. A cold iron requires more time in contact with the board, and makes it harder to get a good joint. So, you are just as likely to damage the board if the iron is too cold, as if it is too hot.
3. We suggest a chisel tip, approximately 2mm wide for the signal and ESC/motor wires. A thin pencil tip will not transfer heat effectively. A heavy bevel tip will be hard to maneuver precisely.
4. A heavy bevel tip may be appropriate for the main battery leads, which are 12 or 14 gauge, but a good quality iron can get that job done with a general purpose chisel tip if you don't have a bevel tip or don't want to change out your tip.
5. Be sure the tip is clean. If the tip has black or grey oxidation on it; if the tip has been damaged by abrasive or chemical cleaning; if the tip has been damaged by overheating; then you are more likely to damage the board. The tip should be shiny silver and the solder should stick to the tip easily and not want to "ball up" on the tip.
6. Use leaded solder. Yes, lead is bad for you, but you don't get significant lead exposure when soldering. Unless you eat the solder. You're not going to eat the solder, right? Okay... then you're fine. Use leaded solder. Lead free solder is much harder to solder with. It's largely intended for commercial use, where keeping lead out of landfills is important.
7. Before soldering the joint, prep the joint and wire. Tin the pad by putting a small amount of solder on it. Then strip the wire, twist the strands together, and tin the wire. Snip the wire to length. Then hold the wire against the pad and touch both the pad and the wire with the hot iron to flow the solder and complete the joint.

If you are interested in spending some time getting soldering tips, I suggest these videos:

- <https://www.youtube.com/watch?v=KPbozW0Y3p4>
- <https://www.youtube.com/playlist?list=PLwoDb7WF6c8kjhbHG8JR2IRnxJMU8pYt>



# Assembly Instructions



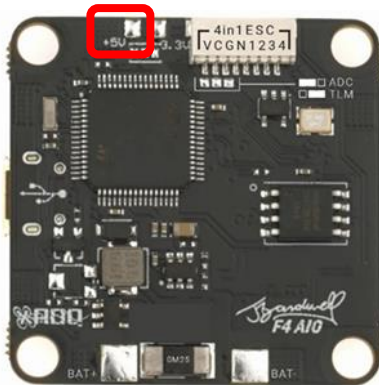
## On the bottom of the board.

If you are using individual ESCs, solder the main battery connector to the large BAT+ and BAT- pads on the JBF4.

If you are using a 4-in-1 ESC, solder the main battery connector to the 4-in-1 ESC, not the JBF4.

The battery connector should use either 12 gauge or 14 gauge, fine-strand, silicone-insulation wire. This type of wire is commonly referred to as "silicone wire". Do not use regular stranded copper wire for the battery lead. It will become brittle over time and break.

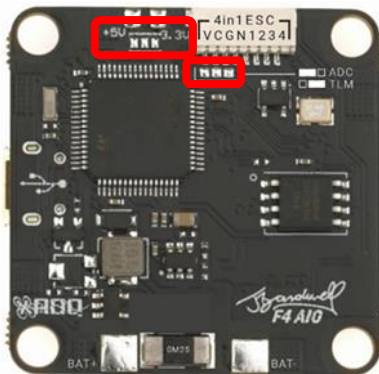
The battery connector will most commonly be an XT60.



## On the bottom of the board.

If you will be using any 5v accessory other than your receiver, you can solder its power and ground wires here. The Buzzer+ pad on top of the board can also be used for 5v power if you prefer that wiring.

Remember that the total power draw of all 5v accessories connected to the board cannot exceed 1.2 amps.



## On the bottom of the board

If you are using a Spektrum Satellite receiver, it probably requires 3.3v input voltage. If you are using any other receiver, it probably requires 5v input voltage. If you're not sure, double check the receiver's specifications online.

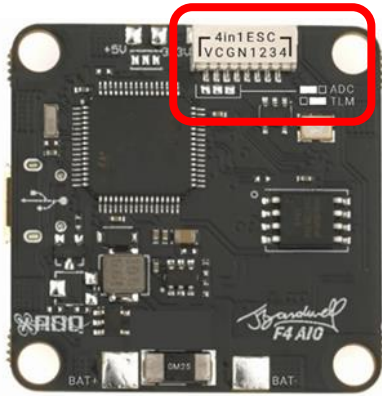
Create a *solder bridge* connecting the center pad of the 5v/3.3v selection pads to either the 5v or the 3.3v outside pad. This will select which voltage the flight controller sends to your receiver.

If you are using the 4-in-1 plug, create a solder bridge on the pads underneath the 4-in-1 plug to select analog current sensing (ADC) vs. ESC Telemetry (TLM). This solder bridge selects the function of the C pin in the 4-in-1 plug.

To create a solder bridge, first tin both pads generously. Then hold your soldering iron in between the two pads, touching both of them at the same time. If they don't immediately flow together, feed a little bit of solder in until it overflows the pad and joins together into a single blob that spans both the center and the outer pad.

If you accidentally bridge all three pads together and then power the board, the board may be damaged. Confirm both visually and with your multimeter that this has not happened before proceeding.

[See an example of how to create a solder bridge here.](#)



### On the bottom of the board

If you are using a 4-in-1 ESC, verify that the ESC's plug is compatible with the JBF4's plug. For Airbot ESCs, the pinout should already match, **but you still must verify this!** For all other ESCs, you will have to build your own wire header. RaceDayQuads sells a pre-crimped wire header set that is perfect. It can be purchased here:

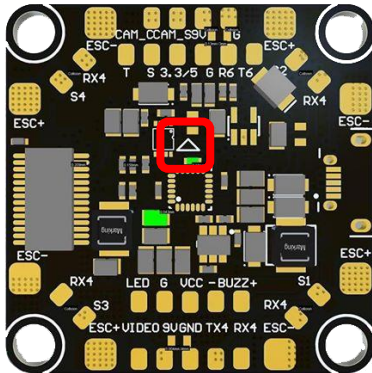
<https://www.racedayquads.com/products/sh-1-0-silicone-cable-set>

V should go to the vBat output on the ESC. It can be connected to the main battery positive pad on the ESC as well.

C should go to the analog current sensor output on the ESC or to the ESC Telemetry wire. Use the solder bridge below the plug to choose which of these functions you want to use. You cannot use both functions at the same time.

G should go to the Ground output on the ESC.

1, 2, 3, 4 should go to the four motor outputs on the ESC.



Before performing the next step, install your motors on your frame. If you are using individual ESCs, install them and solder them to the motors.

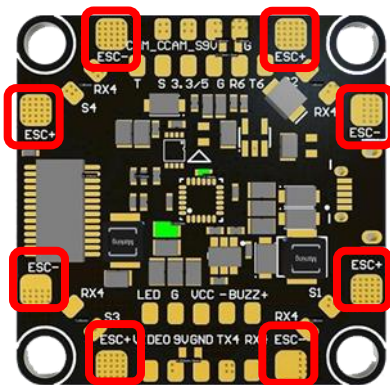
If you are using a 4-in-1 ESC, install it in the frame and solder the motors to it. Connect the JBF4 to the ESC using the plug socket on the bottom of the JBF4. You have just saved yourself a LOT of soldering.

Install the flight controller in your quadcopter frame. Do not completely reassemble the frame. Leave the flight controller accessible so that you can solder wires to it. For example, leave the top plate of your frame off, or remove the camera pod of an X-style frame so that you are working only with the base-plate.

Having the FC mounted in the frame will allow you to ensure that wire lengths are correct as you solder accessories to the board. When in doubt, it's always better to leave a little extra wire.

Be 100% sure that the front-facing arrow on the JBF4 faces the front of the quadcopter! If for some reason this is not possible, then you must use Betaflight's "board align" feature to compensate.

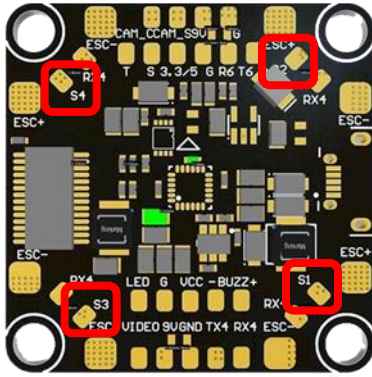
[Learn more about the "board align" feature here.](#)



Solder the ESC power wires to the ESC+ and ESC- pads at the corners of the flight controller. ESC power wires are typically 18 or 20 gauge. Some ESCs are sold without attached wires, in which case you will need to provide your own. We recommend 20-gauge silicone wire for this purpose.

Each ESC should be soldered to the pads in the corner closest to it. Cut the wires short to eliminate excess wire that can get caught up in props, but be sure to leave yourself a little slack. You can always shorten a wire a little more if you need to, but you can't easily make it longer again.

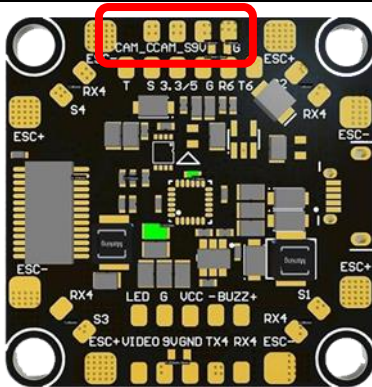
Be absolutely sure that you do not mistakenly reverse the wires. Red goes to ESC+ and black goes to ESC-.



If you are using a 4-in-1 ESC, skip this step. Your signal wires are already in the plug that you made.

If you are using individual ESCs, solder the ESC signal wires to the S1, S2, S3, and S4 pads. The signal wire is a thin wire, typically yellow or white. It's usually connected to the same side of the ESC as the power wires.

Some ESCs have a signal ground wire and some ESCs don't. If your ESC has a thin black wire that is twisted with the signal wire, that is the signal ground. Solder that wire to the ESC- pad next to each of the S1, S2, S3, and S4 pads. You may also simply remove that wire from the ESC.



Solder the FPV camera's video wire to the CAM\_S pad. This wire is typically yellow.

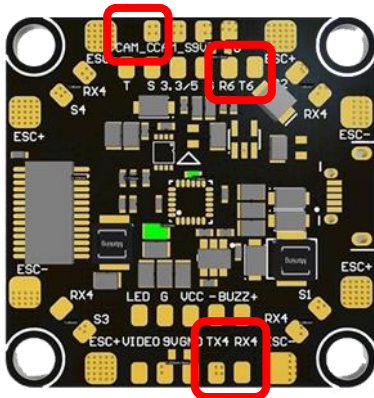
Solder the FPV camera's power wire to the 9V pad that is next to the CAM\_S pad. This wire is typically red. The back of the camera will also typically have a pinout label for the plug.

Remember that the accessories wired to the 9v regulator cannot exceed 1.2 amps current draw in total.

Solder the FPV camera's ground wire to the G pad that is adjacent to the 9V pad. This wire is typically black.

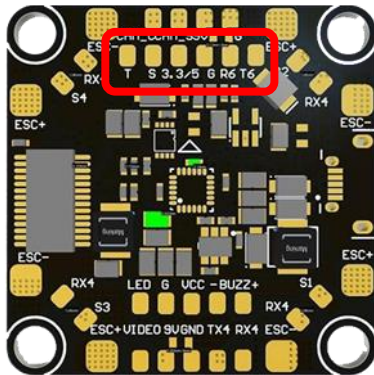
FPV Camera Control is a feature of Betaflight that allows you to access your FPV camera's on-screen menu using your transmitter sticks. With this feature, you can easily adjust brightness, contrast, and other camera settings in response to changing lighting conditions.

There are two forms of Camera Control: analog and digital. Analog camera control is for cameras that use an analog-style joystick input. Digital camera control is currently reserved for some Runcam cameras. Look at the plug on the back of your camera. If you see a pin labeled "OSD" then your camera uses analog control. If you have a Runcam camera with pins labeled "RX" and "TX", then your camera uses digital control.



If your camera uses analog control, solder the “OSD” wire on the camera plug to the CAM\_C pad on the JBF4.

If your camera uses digital control, solder the camera TX wire to pad R6 or RX4 on the JBF (whichever UART you decided to use in the Pre-Installation Planning Sheet). Solder the camera RX wire to pad T6 or T4 on the JBF4.



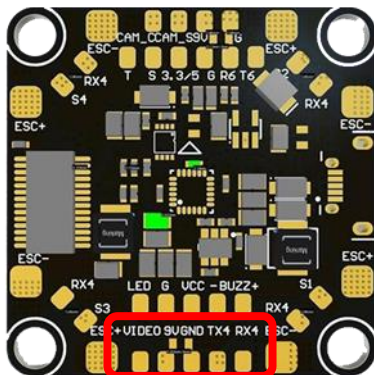
You must use a receiver that supports a serial protocol such as SBUS, iBus, and SpekSat. **You cannot use a PPM or PWM receiver with the JBF4 AIO.**

Solder the receiver signal wire(s) to the UART pad(s) that you determined in the Pre-Build Worksheet.

Solder the receiver ground wire to the G pad closest to the UART pad. In a typical cable, the ground wire will be black or brown.

If you are using a FrSky receiver with SmartPort telemetry, solder the SmartPort wire to the T pad. If you are using any other type of telemetry, solder the wire to the UART pad that you determined in the pre-build worksheet.

Solder the receiver power wire to the 3.3/5 pad. If you are using a 3.3v receiver, it would be prudent to double-check with a multimeter that 3.3v or less is coming out of this pad. Do not connect a receiver that takes 3.3v power to a 5v pad or you will destroy it.

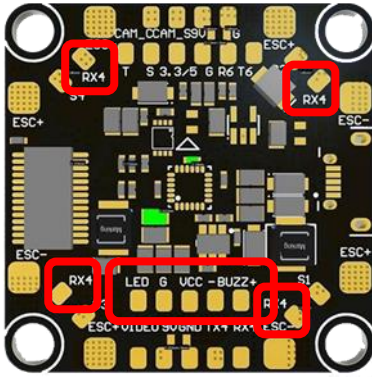


Solder the FPV video transmitter’s video wire to the VIDEO pad. This wire is typically yellow.

Solder the FPV video transmitter’s power wire to the 9V pad that is next to the VIDEO pad. This wire is typically red. Remember that the accessories wired to the 9v regulator cannot exceed 1.2 amps current draw in total.

Solder the FPV video transmitter’s ground wire to the GND pad that is adjacent to the 9V pad. This wire is typically black.

If you are using vTX Remote Control (SmartAudio, Tramp Telemetry), solder your vTX’s control wire to the pad that you determined in the Pre-Build Worksheet. For most SmartAudio devices, the control wire will be the audio wire. For Tramp Telemetry, there is typically a dedicated wire separate from the audio wire.



If you intend to install a buzzer on your quad, solder its positive and negative legs to the BUZZ+ and BUZZ- pads. Most likely, you should not solder the legs directly to the pads. You should use a short length of silicone wire, thickness between 30 gauge and 26 gauge, and then mount the buzzer somewhere secure on the quad.

If you intend to use ESC telemetry, then, for individual ESCs, solder the ESC Telemetry wire from the ESC to the RX4 pad at each corner of the JBF4. For a 4-in-1 ESC, there will be only one ESC Telemetry wire. Solder it to the UART pad that you determined in the pre-build worksheet.

If you are using programmable LEDs, solder their signal wire to the LED pad. Solder their ground wire to the G pad that is right next to the LED pad. Solder their power wire to an appropriate location, depending on what voltage the LEDs require. The simplest scenario is to get LEDs that can take your battery voltage, and then solder the LED power wire to the VCC pad.

# Updating Betaflight Firmware

---

Like all software, the software that runs your flight controller has versions. Just like Windows XP was followed by 2k, then 7, 8, and 10. The software that runs your flight controller is called Betaflight. Putting a new version of Betaflight on your JBF4 AIO is called “flashing” your board.

Even if you decide you don't want to update your firmware right now, you still need to install the VCP driver to configure the board. So, you must at least go through step 1 below to use your board.

## Installing Drivers

Before you can flash your board, you must get the drivers installed on your computer. If you are on MacOS or Linux, Your O/S comes with the drivers you need pre-installed. You do not need to perform this step. You may skip directly to “Installing Betaflight Configurator”.

If you are on Windows, you must install the drivers manually. This is a place where beginners often struggle. We’re going to present the steps here, and if you can follow them, then great, but many people won’t be able to follow them. If you need a more personal approach, [here is a link to a video that walks you through the process in more detail.](#)

The video linked above shows a process of using Zadig to replace the VCP driver. The ImpluseRC Driver Fixer described below is an easier way of doing the same thing. So, use the ImpulseRC Driver Fixer and don’t mess around with Zadig like the video shows.

Here are the steps you will perform:

8. Download the Virtual COM Port (VCP) driver installer from [here](#). Unfortunately, you must either create a login or give them your email address, then wait for them to email you a link, to download the actual installer. We recommend that you create a user login because the “send you a link in an email” method sometimes takes a long time to go through.
  - a. To create a login at the STM site, first go to [this URL](#).
  - b. Click “Create an Account”
  - c. Enter your information in the next page and click “Register”
  - d. You will receive a confirmation email. Complete the confirmation process.
  - e. Log in to the STM site using the login you just created and download the VCP driver installer from the link in step 1.
9. Run the VCP installer and let it finish.
10. Download the ImpulseRC Driver Fixer from [here](#).
11. Run the ImpulseRC Driver Fixer. It will instruct you to plug in your flight controller.
12. Plug the flight controller into your PC via USB. The ImpulseRC Driver Fixer should complete successfully.

## Installing Betaflight Configurator

Betaflight is managed using the Betaflight Configurator application, also known for short as the Betaflight GUI, or just, “The GUI”. (GUI is pronounced “gooey” in case you wondered. Only weirdos say, “Gee You Eye”.)

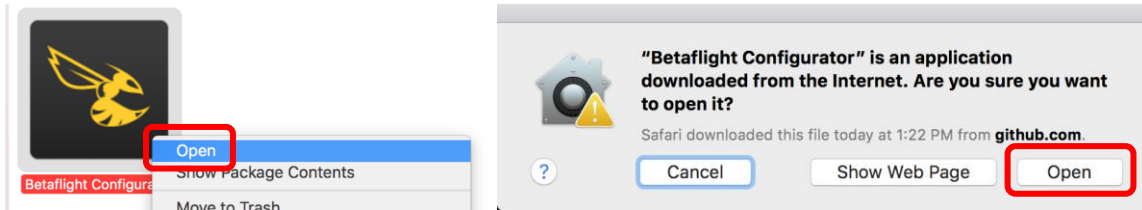
Betaflight used to be a Chrome App until literally the day before this manual released, and then Google killed all Chrome apps, so now it’s 12:15 AM and I am re-writing this section. And I’m leaving this paragraph in too! If you’re wondering where the Chrome App went, that’s where.

You can download the Betaflight Configurator from here:

<https://github.com/betaflight/betaflight-configurator/releases/>

This is a standard application package for Windows, MacOS, or Linux. After you download it, install and run it the same as any other application.

On MacOS you must give the application permission to run and access the network. The simplest way to do this is to right-click the installer and choose Open. A security prompt will appear asking, "Are you sure you want to open it?" Click Open on this prompt and the relevant configuration changes will be made automatically.



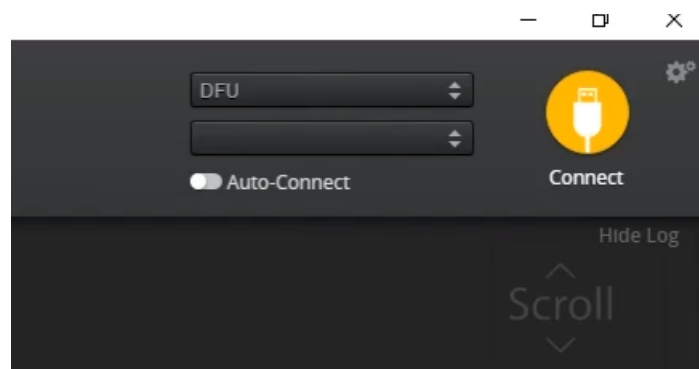
## Flashing New Firmware

At this point, if you want to update your firmware, here is how to do it. But if you are annoyed and frustrated at this point and just want to go fly, please, go for it! You don't have to be running the absolute latest firmware to have a good time. Just forget about this nonsense!

Still here?

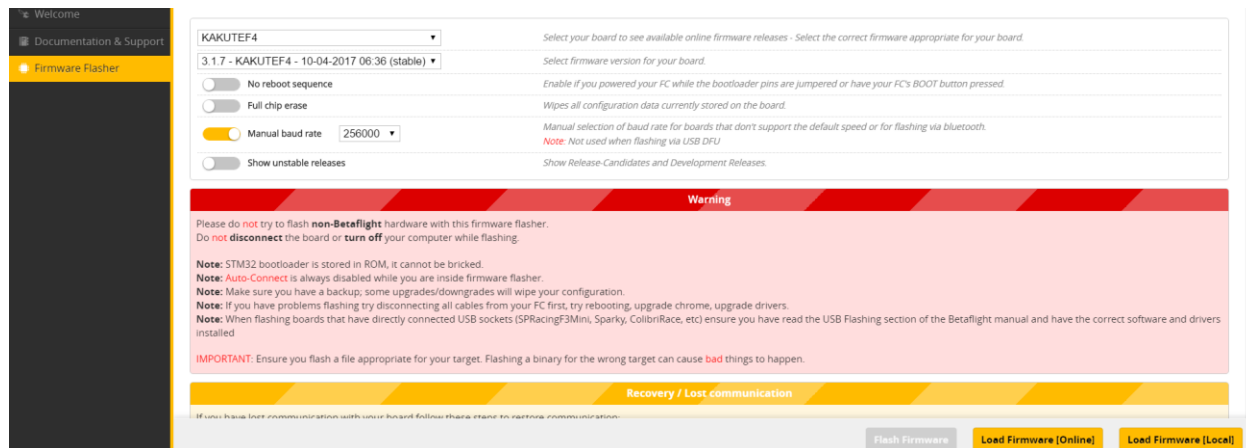
To flash your firmware, you must connect the board in "bootloader mode". Bootloader mode means that the board is ready to accept new programming. To put the board into bootloader mode, hold down the bootloader button while plugging in the USB cable. Leave the button pressed for a moment after plugging in the USB cable to be sure it "takes".

If your JBF4 AIO is in bootloader mode, then you will see "DFU" in the pulldown menu in the upper-right of the configurator, as shown here:



If you don't see DFU in the pulldown menu, then either the board didn't detect that you had the bootloader button pressed or your drivers are not installed correctly. If you don't see DFU in the pulldown menu, you cannot flash new firmware to the board. It won't work. The fix is usually to re-run the ImpulseRC Driver Fixer app described above.





Here are the remaining steps:

1. Go to the “Firmware Flasher” tab.
2. Select “CLRACINGF4” in the “Choose a board” pulldown menu. If you flash any other board type, the JBF4 AIO will not function. It won’t be damaged, it just won’t work until you flash CLRACINGF4 to the board.
3. Select the latest version of Betaflight in the “Choose a firmware version” pulldown menu.
4. Click the “Load Firmware (Online)” button. The button will change to read “Downloading”. The Flash Firmware button will change from gray to orange.
5. Click the “Flash Firmware” button. The screen should automatically scroll down to show the status bar, which will fill with orange as the flashing process completes.
6. Flashing will be followed by a process called “Verifying”. Verification sometimes fails, but this isn’t a problem. If the flash completes, everything is usually fine.
7. Un-plug your board and then plug it back in again, this time without holding down the bootloader button.
8. For Windows users, the pulldown menu in the upper right of the configurator will read COM3 (or some other number). For MacOS and Linux users, the pulldown will read something starting with /dev/tty. This is normal. If the pulldown menu reads “Manual Selection” then your board is not being detected. This may indicate that you did not use the CLRACINGF4 target when you flashed the board. Or it may indicate that your Virtual COM Port (VCP) drivers were not installed correctly.

You are ready to configure your board.

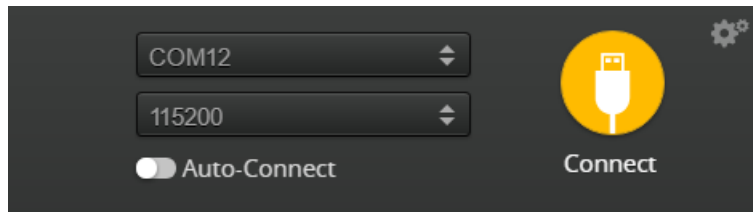
# Initial Configuration

---

The full configuration of Betaflight could take hours to document. In this section, we'll describe a few things that are specific to this board.

## Connect to The Board

Plug the board in to USB. Start Betaflight Configurator. You should see "COM3" (or some other number) in the upper-right menu. Click "Connect".



The GUI will load.

## Ports

Click on the “Ports” tab on the left side of the window.

Ports

WIKI

**Note:** not all combinations are valid. When the flight controller firmware detects this the serial port configuration will be reset.  
**Note:** Do **NOT** disable MSP on the first serial port unless you know what you are doing. You may have to reflash and erase your configuration if you do.

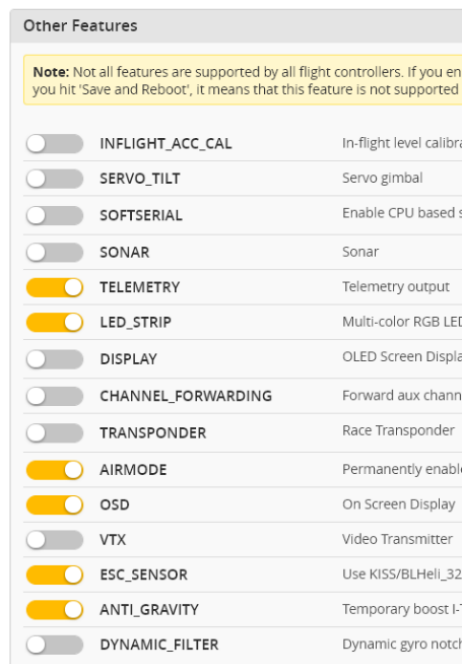
Identifier	Configuration/MSP	Serial Rx	Telemetry Output	Sensor Input	Peripherals
USB VCP	<input checked="" type="checkbox"/> 115200 ▾	<input type="checkbox"/>	Disabled ▾ AUTO ▾	Disabled ▾ AUTO ▾	Disabled ▾ AUTO ▾
UART1	<input type="checkbox"/> 115200 ▾	<input checked="" type="checkbox"/>	Disabled ▾ AUTO ▾	Disabled ▾ AUTO ▾	Disabled ▾ AUTO ▾
UART3	<input type="checkbox"/> 115200 ▾	<input type="checkbox"/>	SmartPort ▾ AUTO ▾	Disabled ▾ AUTO ▾	Disabled ▾ AUTO ▾
UART4	<input type="checkbox"/> 115200 ▾	<input type="checkbox"/>	Disabled ▾ AUTO ▾	Disabled ▾ AUTO ▾	TBS SmartAudio ▾ AUTO ▾
UART6	<input type="checkbox"/> 115200 ▾	<input type="checkbox"/>	Disabled ▾ AUTO ▾	ESC ▾ AUTO ▾	Disabled ▾ AUTO ▾

- USB VCP is the port that is used to talk between the JBF4 AIO and the Configurator GUI. MSP is the protocol that they use to talk to each other. If you ever turn off MSP on USB VCP, you will not be able to contact the board any more via the Configurator. You will have to re-flash the board and erase your configuration to get back into it. Suffice it to say, this is bad, and you shouldn't do it. **Don't ever turn off MSP on the USB VCP line in the Ports tab.**
- Enable the Serial RX option on whichever UART you used for your receiver.
- If you are using FrSky SmartPort telemetry: on the UART3 line, in the Telemetry Output column, choose SmartPort from the pulldown menu. If you are using any other kind of telemetry, select the appropriate protocol on the UART that you connected the telemetry wire to.
- If you are using vTX Remote Control, to control your video transmitter, on the UART4 or UART6 line (whichever UART you used), in the Peripherals column, choose TBS SmartAudio or IRC Telemetry (whichever your vTX uses) from the pulldown menu.
- If you are using ESC Telemetry, in the Sensor Input column, choose ESC for the UART4 line. If you are using a 4-in-1 ESC and want to get current information from it, go to the Power And Battery tab and select “ESC Sensor” as your Current and Voltage source. This will cause the FC to take current and voltage information from the ESC sensors instead of its own built-in sensors. You must be using DSHOT in order to use ESC Telemetry.

## Configuration

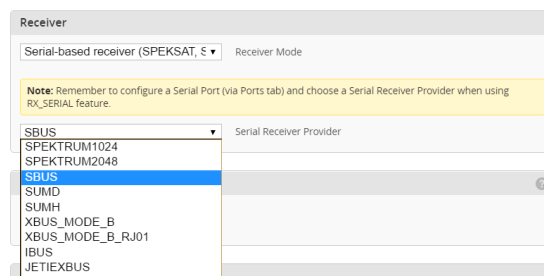
Click on the “Configuration” tab on the right side of the window. Scroll down to the “Other Features” section.

- If you are using telemetry (such as SmartPort), enable TELEMETRY. **If you are using SmartPort, you must if you are using SmartPort Telemetry wired to the TELE pad, you must go into the CLI and type, “set tlm\_halfduplex = off”, then, “save”.**
- If you are using a programmable LED strip, enable LED\_STRIP.
- Air Mode increases authority when the throttle is all the way down. Joshua flies with this option on all the time but some pilots don't like it.
- The JBF4 AIO has built-in Betaflight OSD. The OSD option should always be enabled.
- If you are using ESCs with telemetry capability, enable the ESC\_SENSOR option.
- The ANTI\_GRAVITY option increases the stability of the copter when the throttle is raised or lowered quickly. We recommend enabling this option, and setting Anti Gravity Gain to 5.0 in the PID Tuning tab. Detailed instructions for tuning this value are outside the scope of this manual, but this default setting will work well for most quads.



Next, go to the Receiver section of the “Configuration” tab. Since the JBF4 AIO only supports serial-type receivers, configuration of this section is simple.

- Set the Receiver Mode to “Serial-based receiver”.
- Set the Serial Receiver Provider to the type that matches what kind of receiver you have. FrSky and other SBUS receiver should use SBUS. Spektrum Satellite receivers should use SPEKTRUM2048 or SPEKTRUM1024 depending on whether they are DSM2 or DSMX. FlySky receivers use iBus. Crossfire receivers should select CRSF.



## Blackbox

If you have enabled the Blackbox feature, go to the Blackbox tab on the left-hand side of the window. In the Blackbox tab, at the top, set the Blackbox Logging Device to SD Card and the Blackbox Logging Rate to 2 kHz.

### Blackbox

#### Blackbox configuration

SD Card ▼ Blackbox logging device

2 kHz ▼ Blackbox logging rate

**Save and reboot**

## OSD

In the OSD tab, you can choose which values you want to see on screen while you are flying. Enable and disable individual elements using the Elements toggles on the left.

The Video Format section lets you choose whether your camera is NTSC or PAL. Betaflight defaults this value to Auto, but Auto sometimes picks wrong. If this happens, you might not see any OSD text, or the bottom of the OSD text might be off the bottom of the screen. In some cases, you won't see any video at all—just the OSD. Because of this, we recommend manually setting the Video Format to NTSC or PAL depending on which type of FPV camera you have.

You can rearrange the individual OSD elements on screen by dragging them with your mouse.

### OSD

WIKI

**Note:** OSD preview may not show the actual font that is installed on the flight controller.


#### Elements

Switch all:

- Rssi Value
- Main Batt Voltage
- Crosshairs
- Artificial Horizon
- Horizon Sidebars
- Timer 1
- Timer 2
- Flymode
- Craft Name
- Throttle Position
- Vx Channel
- Current Draw
- Mah Drawn

#### Preview (drag to change position)

Logo:



#### Video Format

AUTO  PAL  NTSC

#### Units

IMPERIAL  METRIC

#### Timers

1 Source: ON TIME ▼  
Precision: SECOND ▼  
Alarm: 10

2 Source: TOTAL ARMED TIME ▼  
Precision: SECOND ▼  
Alarm: 10

Some individual OSD Elements of note are:

**Main Batt Voltage:** This is probably the single most important element to use. Your battery must never go below 3.3 volts per cell under any circumstances, or it will suffer damage. For a 3S battery, this would be 3 times 3.3 = 9.9 volts. For a 4S battery, it would be 13.2 volts. Any time the battery goes below this level, it is likely being damaged, at least a little. More importantly, you will have very little power and may crash if you are doing aggressive maneuvers.

Although 3.3 volts is the recommended absolute minimum, you will get the best life out of your batteries if you keep the voltage above about 3.5 volts per cell—10.5 volts for 3S and 14.0 volts for 4S. If you are flying and you notice the battery going below this level, it might be time to land, or at least go easier on the throttle. If your batteries consistently drop to this level when you try to fly, it might be time to get new batteries.

When you are using the battery, its voltage will drop. This is referred to as “sag”. After you land, the battery will recover a small amount and its voltage will rise. Our recommendation is that, the battery should be at no less than 3.75 volts per cell—11.25 volts for a 3S and 15.0 volts for a 4S—after it has been allowed to rest at the end of a flight. If your batteries are consistently resting at below this level at the end of a day of flying, then you might be shortening their lifespan at least a little.

**mAh Drawn:** Although voltage is what ultimately determines whether a battery is being damaged, mAh may be a better way of deciding when to land. Because voltage sags when you raise the throttle and recovers when you lower the throttle, it can be hard to tell exactly how used-up the battery is. mAh is like a “gas gauge” for your battery. It shows how much capacity you have consumed, independent of other factors.

A good practice is to draw from a battery no more than about 80% of its rated mAh. So, a 1300 mAh battery would be able to deliver about 1000 to 1100 mAh. With a healthy battery and an honest battery rating, this will result in the battery resting at about 15.0 volts after flying. If you draw 80% of a battery’s rated capacity and it is resting at higher than 15.0 volts, the manufacturer might have under-rated its capacity. If the battery is resting at lower than 15.0 volts, the manufacturer might have over-rated its capacity. Also, remember that batteries lose capacity as they age. A battery that consistently fails to deliver 80% of its rated capacity without sagging below 14.0 volts might need retirement.

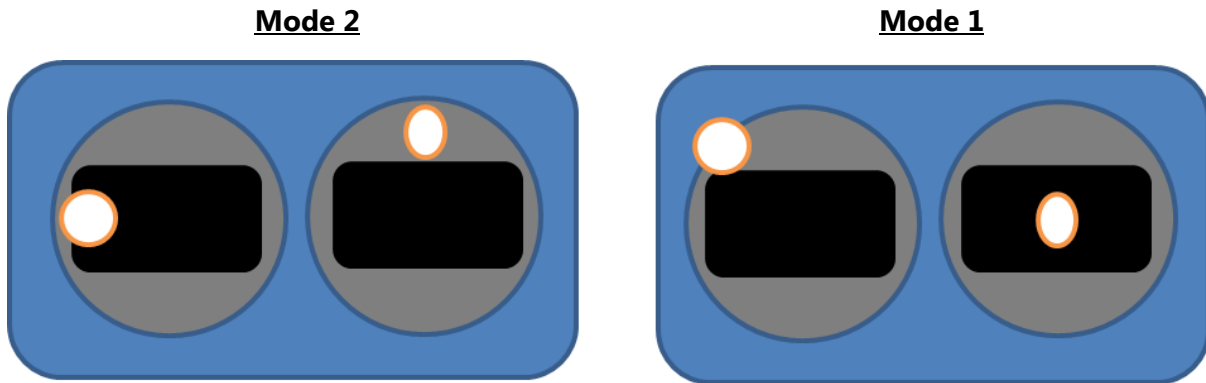
The mAh readout is based on the current sensor on your JBF4 AIO board. In v2 of the JBF4, the current sensor has been calibrated from the factory and should be accurate simply by setting the Amperage Meter scale to 250. This only applies when using the current sensor on the JBF4, such as if you have soldered the XT60 battery leads directly to the pads on the JBF4. If you are using a 4-in-1 ESC with current sensing, you will need to calibrate it manually.

The screenshot displays the configuration interface for the JBF4 AIO board, organized into three main sections: Battery, Voltage Meter, and Amperage Meter.

- Battery Section:** Contains dropdown menus for "Voltage Meter Source" and "Current Meter Source", both set to "Onboard ADC". Below these are four numeric input fields: "Minimum Cell Voltage" (3.3), "Maximum Cell Voltage" (4.3), "Warning Cell Voltage" (3.5), and "Capacity (mAh)" (0).
- Voltage Meter Section:** Shows a "Battery" readout of "0.1 V". It includes three numeric input fields: "Scale" (110), "Divider Value" (10), and "Multiplier Value" (1).
- Amperage Meter Section:** Shows a "Battery" readout of "0.00 A". It includes two numeric input fields: "Scale" (250, highlighted with a red box) and "Offset in milliamperere steps" (0).

# Using The OSD

If you are using a Betaflight Flight Controller with Betaflight OSD, you can manage the vTX transmit power and channel from within the OSD. The vTX must support this function.

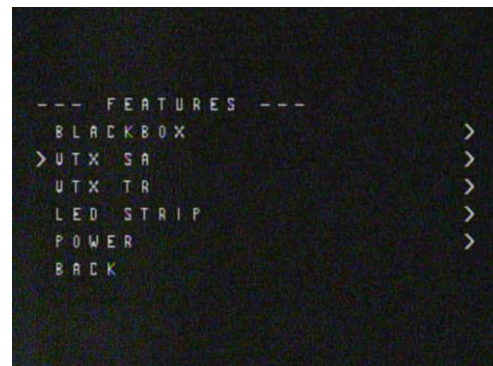


The graphics above show the stick command to bring up the OSD menu. The stick command is: throttle centered, yaw left, pitch forward. The exact stick command therefore depends on which mode your transmitter sticks are in.

In the OSD menu, use pitch up/down to move the cursor between menu items. When a menu option has a > symbol to the right of it, this indicates that it contains a sub-menu. Roll-right will enter the sub-menu. For example, in the screen to the right, moving the cursor to "Features" and then moving the roll stick to the right will enter the "Features" sub-menu.



To manage the vTX, enter the "Features" menu. From there, enter "VTX SA" if you are using a SmartAudio device, or "VTX TR" if you are using an ImmersionRC Telemetry device.





The screen to the right shows the current vTX settings. From here, you can change the frequency band, channel, and power level of the video transmitter. After making the changes, move the cursor to "Set" and press roll-right to confirm the settings.



# Using FPV Camera Control

---

If you have connected the CAM pad on the JBF4 to your FPV camera, or if you are using a Runcam camera with TX and RX pads, then you can use the FPV Camera Control feature of Betaflight. This feature allows you to access your FPV camera's setup menu using your transmitter sticks.

Here is how to use the Camera Control feature. With the quad disarmed, put the throttle at 50% (centered). Push yaw-right on the transmitter stick. You are now in Camera Control mode.

When in Camera Control mode, the pitch and roll inputs (right stick for Mode 2 transmitters) input "up/down/left/right" presses on the camera joystick. Pushing yaw-right inputs one "click" on the camera joystick. Use these inputs exactly as if you had plugged a joystick into the back of the camera.

When you are done adjusting the camera, push yaw-left to exit Camera Control mode.

If you are using the CAM pad on the JBF4, the camera control function may not work with every FPV camera. This is a limitation of how Betaflight implements this function, and not a defect in the JBF4. After setting up Camera Control, we recommend that you center-throttle & yaw-right two or three times in a row. If you are using digital control, the menu should appear. If you are using analog control and the menu does not appear, your camera may not work with this feature, or you may need to adjust some command-line parameters. See the Troubleshooting section of this manual for specific instructions.

# Saving Your Configuration

---

Once you have finished building, configuring, and tuning your multicopter, it's a good idea to back up your configuration so that you can restore it later. This is useful if you lose your quad, or if you damage your flight controller, or if you accidentally lock yourself out of your flight controller and must reset it to get back in.

Before we show you the right way to save and restore your configuration, let us warn you about the wrong way. Betaflight and Cleanflight have a "save configuration" and "restore configuration" button. Without going into too much detail, they have some significant drawbacks that mean we don't recommend that you use them.

The correct way to save your configuration is as follows.

- Connect your JBF4 to your PC by plugging in USB.
- Start the Betaflight GUI app.
- Go to the CLI tab.
- In the text box at the bottom of the CLI, type "diff all" and hit enter. This will cause the flight controller to display all configuration options that you have changed from the default values.
- In the lower-right corner of the configurator, click the "Save to File" button.
- Save the file somewhere you won't lose it.

To restore your configuration, do this:

- Open the text file in your text editor.
- Highlight the entire contents of the file.
- Right-click in the text editor window and choose "Copy".
- Connect your JBF4 to your PC by plugging in USB
- Start the Betaflight GUI app.
- Go to the CLI tab.
- Click the mouse once in the text box at the bottom of the CLI tab to place the cursor there.
- Instead of typing any commands, right-click in the text box and choose "Paste".
- Press the Enter key on your keyboard. The pasted-in text will rapidly scroll past.
- Type "save" in the text box at the bottom of the screen.
- Press Enter. The flight controller will reboot and the configuration will be restored.

## Additional Reference

---

Here are some links to additional videos to help you build your quadcopter successfully.



---

*Betaflight 3.3 Ultimate Setup Guide*

---

<https://www.youtube.com/watch?v=8vJCrHj9s6s>



---

*How to Calibrate Your ESCs*

---

<https://www.youtube.com/watch?v=o3Mg-9M0l24>

If you are using an analog protocol like Oneshot or Multishot, calibrating your ESCs is mandatory. Most ESCs today support Dshot. If your ESCs support Dshot, you should use it, and you can skip this step.



---

*Failsafe*

---

<https://www.youtube.com/watch?v=dikr9oDzQqc>

Failsafe is what happens when you fly too far away or your receiver gets disconnected from your flight controller. If you don't configure failsafe, the quad could fly away, or it could crash into something... or someone.

**DON'T FLY WITHOUT VERIFYING THAT FAILSAFE WORKS CORRECTLY.**

To verify failsafe, remove your props, then plug in your battery and arm your copter. Then turn off your transmitter. Your motors should stop within 1 to 3 seconds at most.

If your motors don't stop when you turn off your transmitter, **DO NOT FLY YOUR MULTIROTOR UNDER ANY CIRCUMSTANCES.**

# Glossary of Terms

---

**Flight Controller (FC)** – The “brain” of the quadcopter. It takes commands from the pilot to know what the quadcopter should be doing. It reads data from sensors like the gyro and accelerometer to know what the quadcopter is doing. Then it sends commands to the ESCs and motors to make the quadcopter do what it should be doing. Modern flight controllers often have many non-flight-related functions built in, such as On-Screen Display (OSD).

**Gyro** – Short for gyroscope. A sensor that detects rotation.

**Accelerometer** – A sensor that detects acceleration. Often shortened to just “acc”. Since gravity manifests as a downwards acceleration, the accelerometer is the main sensor that can tell whether the quad is right-side up or not. The accelerometer is required for autolevel flight modes.

**Receiver** – Receives RF signals from the transmitter. Translates them into a protocol that the FC can understand and forwards them to the FC.

**SBUS, iBus, SpekSat** – These are all protocols that different receivers use to communicate the pilot’s commands to the FC. SBUS is used by FrSky devices primarily. iBus is used by FlySky devices. SpekSat is used by Spektrum satellite receivers. FrSky, FlySky, and Spektrum are three different brands or families of transmitter and receiver.

**Transmitter** – Takes control inputs from the pilot via sticks, switches, and so forth. Converts those control inputs to RF signals and transmits them to the receiver.

**Telemetry** – Allows the receiver to send data about the aircraft back to the transmitter, instead of only receiving commands from the transmitter. Example telemetry data might include battery voltage, battery capacity consumed, motor temperature and RPM, GPS coordinates, and so forth. When the receiver and transmitter support telemetry, it is possible to configure alarms on the transmitter, so that the pilot can know about the aircraft’s status (e.g. “battery is low!”) and respond to it (e.g. “land now!”). Most RC control systems support telemetry today, however not all receivers within a system support it.

**ESC** – Short for Electronic Speed Controller. Its function is to make the motors spin. Each motor requires its own ESC. The ESC receives throttle position commands from the FC and makes the motors speed up or slow down to match the commanded throttle position. ESCs may or may not support “ESC telemetry”, a feature where the ESC can send data about the motor to the FC.

**UART** – A serial interface on the FC that is used to communicate with peripheral devices such as the receiver.

**vTX (Video Transmitter)** – Receivers video signal from the FPV camera and transmits it wirelessly into the air.

**vTX Remote Control (SmartAudio, Tramp Telemetry)** – Allows the FC to change the channel, transmit power, and other settings on the vTX, without the pilot having to touch the vTX directly. This allows the pilot to configure the vTX indirectly, such as through the OSD or the transmitter. SmartAudio and Tramp Telemetry are two different protocols that the FC can use to talk to the vTX. Which one you use will depend on which vTX you have purchased.

**OSD** – Short for “On Screen Display”. Displays important data about the aircraft directly on top of the FPV feed, for the pilot to view in the goggles. OSD is built into most modern flight controllers. The data communicated by the OSD overlaps with that communicated by telemetry, although OSD may provide additional visual information that telemetry does not (e.g. artificial horizon).

**FPV Camera** – A camera, typically very small, mounted to the front of the quad. Sends a picture to the vTX so that the pilot can see in their goggles what the quad is doing. The JBF4 AIO supports the FPV Camera OSD Menu Remote Control feature, which allows the pilot to adjust the camera’s menu settings via the transmitter sticks, without plugging a joystick into the back of the camera.

**LED, Programmable** – One or more LEDs that can be commanded to change brightness and color by communicating over a digital serial interface. The JBF4 AIO can communicate with programmable LEDs to customize their appearance in response to quadcopter movement, battery status, and other conditions.

# Faq / Troubleshooting

---

## Smartport telemetry wired correctly and still not working

If you are using SmartPort Telemetry wired to the TELE pad, you must go into the CLI and type:

```
set tlm_halfduplex = off
save
```

If you are using SmartPort wired to a softserial resource, this step is not necessary.

## Receiver Doesn't Work or drops out

One feature that we absolutely required on the JBF4 was the ability to power up the receiver when the FC was plugged in to USB. This lets you test your transmitter settings, flight modes, and so forth without having to plug in a LiPo.

But the USB port can only provide a little bit of power, so we only want the receiver to power up from USB, not any other 5v accessories that you might have connected, such as an LED strip. This is accomplished by installing a diode between the receiver power pad and the 5v regulator.

The diode causes the voltage on the receiver power pad to be slightly lower. The 5v regulator outputs about 5.1 volts; the receiver power pad outputs about 4.8 volts. Most receivers work down to 4.0 volts, but some people have reported that their receiver would not work reliably on the 4.8 volts from the receiver power pad. In this case, the receiver appears to power up and bind, but it does not send signal to the FC. Sometimes it works correctly but drops out or is unreliable.

If you have one of these receivers, the workaround is to solder the receiver's power wire to the 5v pad that is adjacent to the LEDstrip pad. This pad outputs the full 5.1 volts of the regulator, but it will not be powered on when the board is plugged in via USB. Alternatively, you can power your receiver from a 9v pad, but check first to be sure your receiver is rated for this voltage. Some receivers have a voltage range of only 4-6 volts; others go as high as 10 volts.

## Camera Control doesn't work

The JBF4 comes from the factory with a camera control pad already set up, and a resistor already installed. However, you should know that this isn't a guarantee that the feature will simply work with every camera. The feature is known to require some manual tweaking on many cameras. If yours works right out of the gate, wonderful!

For full details, please visit the [Setting Up section](#) of the camera control wiki page. In short, here is what you should try if it doesn't work right off the bat:

- Increase the camera control key delay by going to the CLI and typing, "set camera\_control\_key\_delay=250" then press enter, then type "save" and press enter. If this results in double-clicks of the joystick, reduce the value.
- Change the camera control internal resistance value by going to the CLI and typing, "set camera\_control\_internal\_resistance=100". Unfortunately, there does not seem to be a methodical way of deriving the correct value for this. People have reported it working with values that seem like they must be incorrect. Valid values range from 100 to 500, so work your way up in increments of 50 and see if one works.
- Change the camera control reference voltage by going to the CLI and typing, "set camera\_control\_ref\_voltage=310". Values from 300 to 340 might produce good results.
- Especially if you are using a micro camera, you might need to install a capacitor on the camera side of the OSD circuit. Instructions for doing this are in the camera control wiki page, linked above.

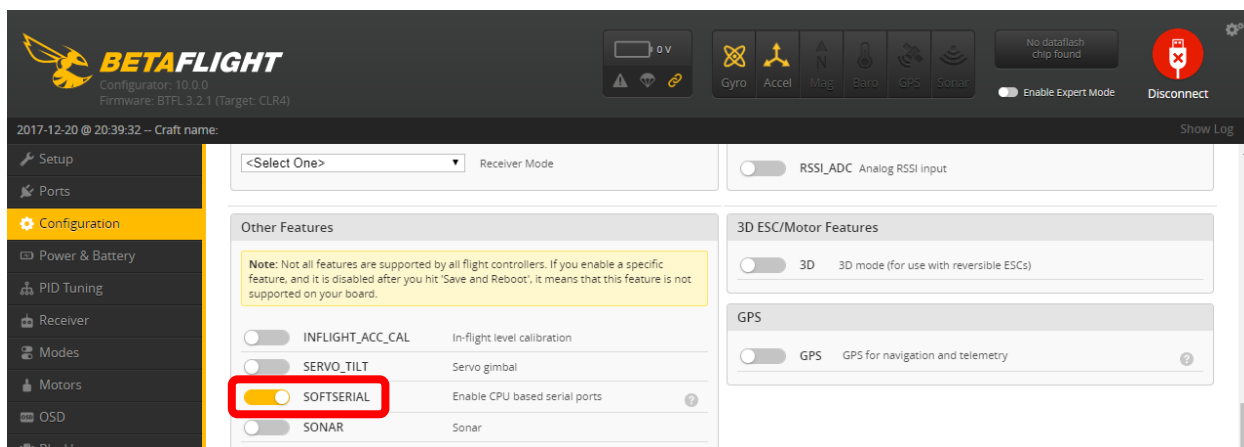


## How To Enable SoftSerial for SmartPort or SmartAudio

If you have a less common configuration, you might run out of available UARTs. One example is if you are using TBS Crossfire and SmartAudio at the same time. In this case, you must use TX4 for Crossfire and not SmartAudio. Another example might be if you damaged the TELEM pad or a UART pad.

Here are the steps for enabling SoftSerial.

1. In the Configuration tab, enable the SoftSerial Feature. Click “Save and Reboot” in the lower-right of the window.



2. In the CLI, type:

```
resource serial_tx 11 B08
resource motor 5 none
resource led_strip none
save
```

3. In the Ports tab, on the SoftSerial1 row, enable either the option in question (SmartPort in the Telemetry Output column or SmartAudio in the Peripherals column). Click “Save and Reboot” in the lower-right of the window.

## Setup Tips for Cleanest Video

The JBF4 is designed so that most builds should not need any additional capacitors or filtering to have clean video. But noise can't be eliminated for every possible setup. Here are tips on how to setup your board for best video quality and the least likelihood of needing capacitors for perfect video.

The JBF4 has a filtered 9v power output for vTX and FPV camera. On paper, this provides the cleanest power supply for cleanest video. However, we have found that some vTX perform better when supplied with direct battery voltage. Although this is counter-intuitive, it's consistent enough that we're passing the recommendation on to you. If you're using the RaceDayQuads Mach 2, ImmersionRC Tramp, or TBS Unfiy HV, we suggest powering from vBat instead of the 9v regulator. Go figure.

We recommend that everyone else start by wiring the JBF4 as shown in the Assembly Instructions section of this manual. If you finish soldering up your JBF4 and have noise, here are our tips:

- Adding a 330 uF, low-ESR, electrolytic capacitor to the power pads of your ESCs will totally eliminate noise in almost all situations. We recommend 25v rated capacitors for 4S, 35v rated capacitors for 5S, and 50v rated capacitors for 6S setups. If you are using a 4-in-1 ESC, add a single 1000 uF capacitor to the main battery power pads. Some 4-in-1 ESCs have separate vBat pads specifically for adding a capacitor; use those if they exist.

If you prefer not to add capacitors, consider one or more of the following options:

- Some vTX have a built-in voltage regulator for powering the camera. It typically outputs 5v. If your vTX can power your FPV camera, then wire your FPV camera directly to the output wire on your vTX, instead of to the power pad on the JBF4.
- If you are one of those purist pilots who doesn't like to use an OSD, you can run the FPV camera's video wire directly to the vTX, instead of using the CAM and vTX pads on the FC.
- Assuming that the FPV camera and vTX can both take the voltage, run both of them off the same power pad on the board. Not just, like, both of them running off of 9v, but literally the same physical pad on the board. Do the same for their ground wires.

If none of the above works, a final step can be to isolate the vTX SMA connector from the frame of the quad. (Assuming you're using an antenna with an SMA connector that is.) In rare cases, the SMA connector touching the frame causes a ground loop that causes horrible video noise.