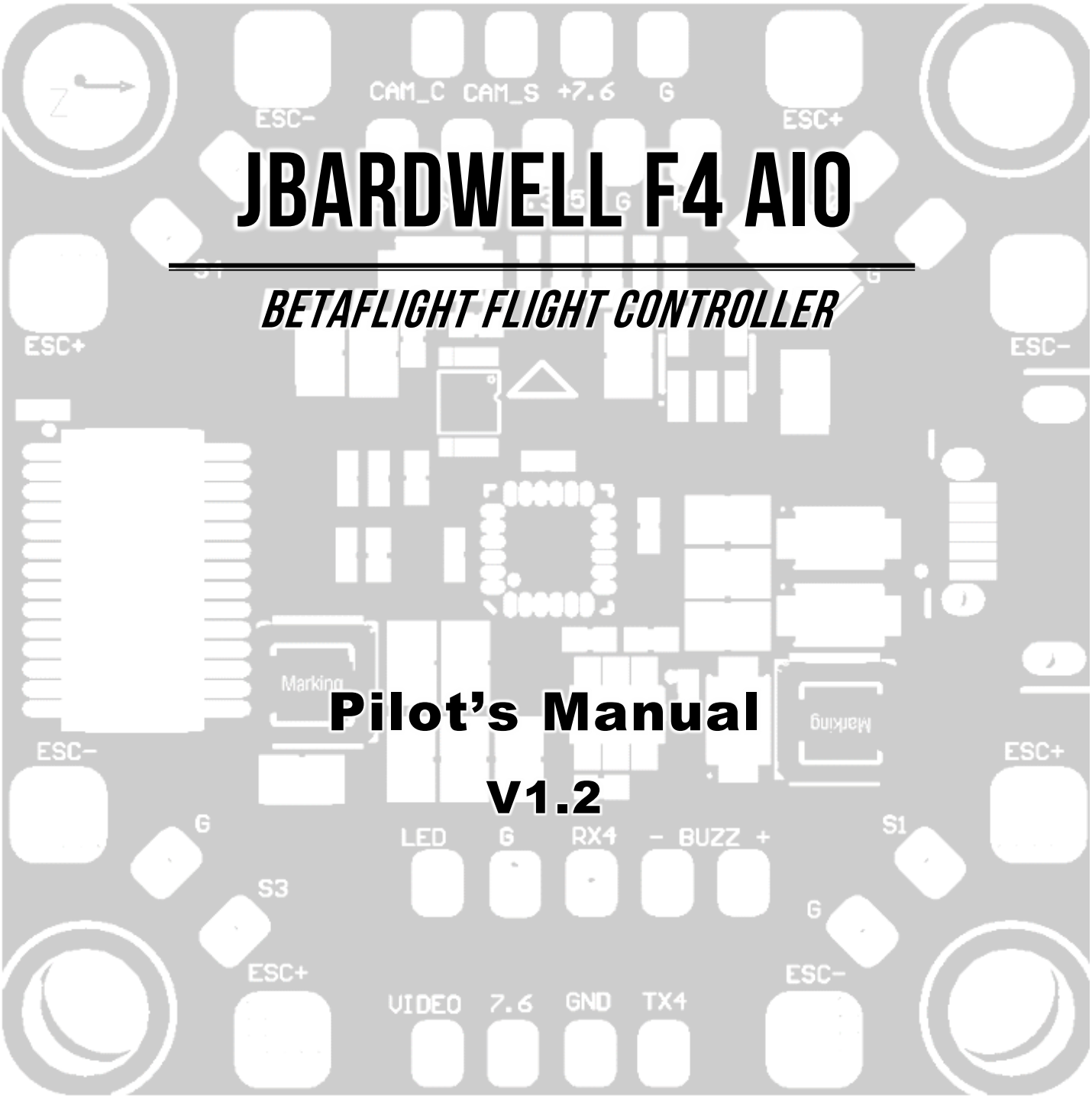
JBARDWELL F4 AIO

BETAFLIGHT FLIGHT CONTROLLER



Pilot's Manual

V1.2

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.

I think it's going to become the hobby's next favorite flight controller.

Input Voltage:	3S to 6S
Processor:	STM32F4
Gyro:	MPU6000
Power Outputs:	3x 5v 1x 3.3v 1x vBat 2x 7.6v filtered
Regulator Rating:	1.2 amps each (5v & 7.6v)
Form Factor:	30.5mm square
Current Sensor:	131 Amps
Betaflight Target:	CLRACINGF4
UARTs:	4 (2 inverted)
Features:	Programmable LEDs SD Card for Blackbox FPV Camera Control SmartAudio Buzzer Output

Manual written, and feature set selected by Joshua Bardwell.

Superb board layout allowing for easy and clean builds.

7.6v filtered output power for vTXs. Lower output voltage increases the filtering of the regulator for the cleanest video.

Centrally-located gyro for reduced noise.

All connections via solder pads for secure connection and high durability.

Dedicated FPV Camera Control pad. Just solder it up. No fussing with resistors and capacitors!

On-board inverter for SBUS and SmartPort protocols. No "uninvert hack" or resource reassignment required.

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. 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 RX6 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	SBUS
iBus, SpekSat, or other non-inverted receiver protocol *iBus telemetry and SpekSat receiver bind mode only supported on RX6*	1 6	SBUS RX6
SmartPort	3	TELE
vTX Remote Control (SmartAudio, Tramp Telemetry)	4	TX4
Programmable LEDs	N/A	LED
ESC Telemetry	6 4 1	RX6 RX4 SBUS
FPV Camera Remote Control	None	CAM_C

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 requires Betaflight's resource remapping feature, which is outside the scope of this 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.
SBUS	Serial receiver (SBUS, iBus, SpekSat) input. UART1. Can be used with both inverted and non-inverted protocols, but iBus telemetry and SpekSat binding don't work on this pad. Can also be used for ESC Telemetry if you're freaky like that.
TELE	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.
+7.6	7.6v filtered output for FPV camera and vTX. Max 1.2 amps.
RX6	UART6 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.
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
TX4 / RX4	UART4 TX and RX pads. TX4 is primarily intended for use with vTX remote control function such as SmartAudio or Tramp Telemetry. UART4 is the only one UART on the board with both TX and RX pads available. If UART4 is needed for some other purpose, reassign LED pad to SoftSerial and use vTX remote control that way.
BUZZ-	Negative leg of piezo buzzer.
BUZZ+	Positive leg of piezo buzzer.
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.
BAT-	Main battery lead negative wire.
+5v/3.3v	Solder bridge for selecting 5v or 3.3v output to receiver pad on top of board.

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.

Resource	UART # / PAD NAME		
Receiver	1/SBUS		6/RX6
SmartPort Telemetry	3/TELEM		
vTX Remote Control		4/TX4	
ESC Telemetry	1/SBUS	4/RX4	6/RX6

We recommend choosing one of the following two configurations. Of course, you're not required to use all of these features. Just ignore the ones you're not using.

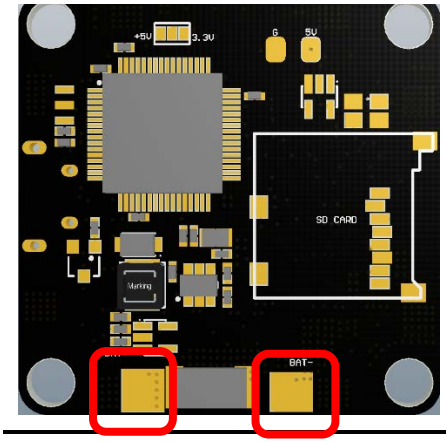
For FrSky SBUS Receiver

Receiver – UART1 / SBUS
SmartPort Telemetry – UART3 / TELEM
vTX Remote Control - UART4 / TX4
ESC Telemetry – UART6 / RX6

For iBus or SpekSat Receiver

Receiver – UART6 / RX6
vTX Remote Control – UART4 / TX4
ESC Telemetry – UART1 / SBUS

ASSEMBLY INSTRUCTIONS

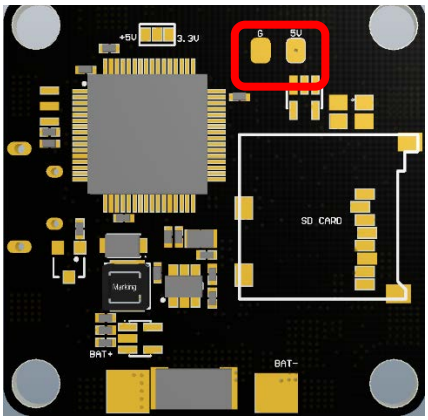


On the bottom of the board.

Solder the main battery connector to the large BAT+ and BAT- pads.

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.

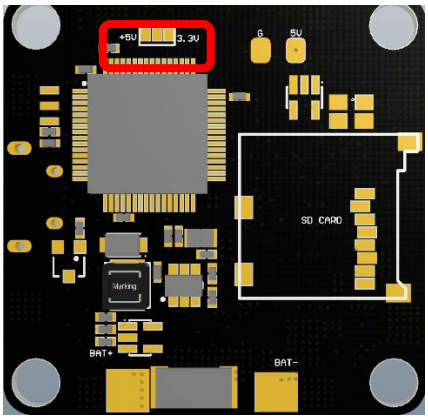
The battery connector will most commonly be an XT60.



On the bottom of the board.

If you will be using any 5v accessory other than your receiver, solder its power and ground wires here.

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

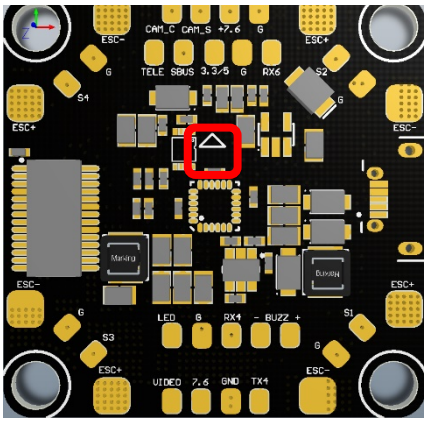
If you are using a Spektrum Satellite receiver, it probably requires 3.3v input voltage. If you are using any other receiver, it probably requires 5v input voltage. If you're not sure, double check the receiver's specifications online.

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.

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.

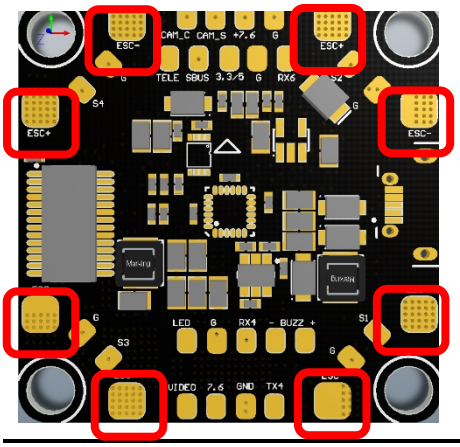


Install the flight controller in your quadcopter frame. Do not completely reassemble the frame. Leave the flight controller accessible so that you can solder wires to it. For example, leave the top plate of your frame off, or remove the camera pod of an X-style frame so that you are working only with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

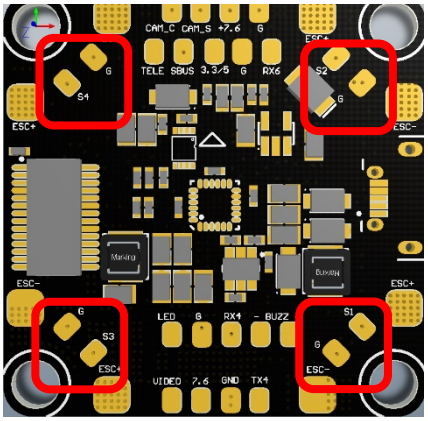
Learn more about the "board align" feature here.



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.

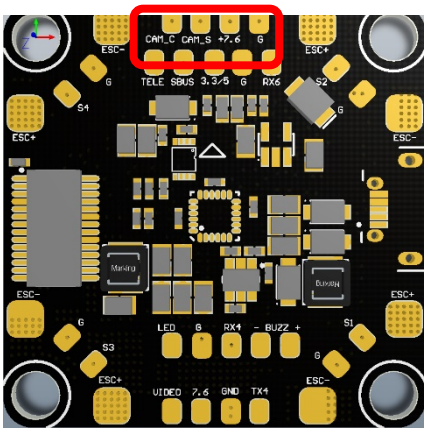
Each ESC should be soldered to the pads in the corner closest to it. Cut the wires short to eliminate excess wire that can get caught up in props, but be sure to leave yourself a little slack. You can always shorten a wire a little more if you need to, but you can't easily make it longer again.

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



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 G pad next to each of the S1, S2, S3, and S4 pads.



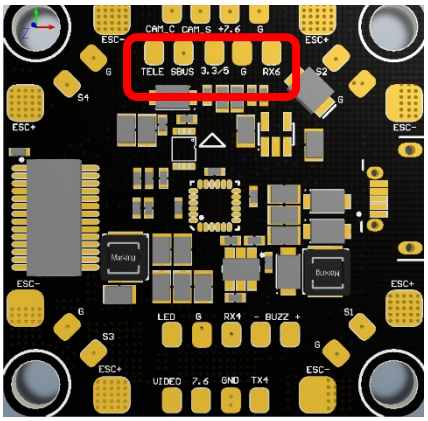
Solder the FPV camera's video wire to the CAM_S pad. This wire is typically yellow.

Solder the FPV camera's power wire to the +7.6 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 7.6v 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 +7.6 pad. This wire is typically black.

If you intend to use Betaflight's FPV Camera OSD Menu Remote Control feature, solder the FPV camera's OSD wire to the CAM_C pad. This wire might be in the same plug as the other wires (Foxeer) or it might be a separate plug (RunCam).



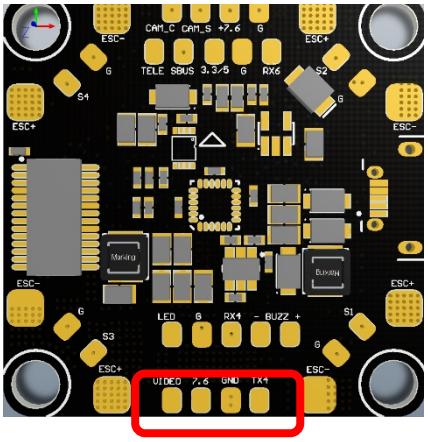
You must use a receiver that supports a serial protocol such as SBUS, iBus, and SpekSat. You cannot use a PPM or PWM receiver with the JBF4 AIO.

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

Solder the receiver ground wire to the G pad next to RX6. 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 TELE pad.

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.

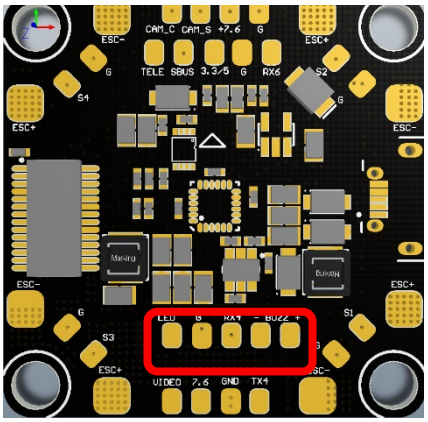


Solder the FPV video transmitter's video wire to the VIDEO pad. This wire is typically yellow.

Solder the FPV video transmitter's power wire to the +7.6 pad that is next to the VIDEO pad. This wire is typically red. Remember that the accessories wired to the 7.6v 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 +7.6 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 your ESCs support telemetry, solder all four of their telemetry wires to the pad that you determined in the Pre-Build Worksheet. Twist the wires together and solder them together before soldering the whole bundle down to the pad.

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 directly to the BAT+ or to any ESC+ 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:

1. 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.
2. Run the VCP installer and let it finish.
3. Download the ImpulseRC Driver Fixer from here.
4. Run the ImpulseRC Driver Fixer. It will instruct you to plug in your flight controller.
5. 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/

Download the ZIP file that matches your O/S: Windows (Win32), MacOS, or Linux. When the ZIP file has downloaded, extract the "BetaFlight Configurator" folder from the ZIP file to a convenient folder on your hard drive, such as your desktop. In the folder, run the Betaflight Configurator app.

FLASHING NEW FIRMWARE

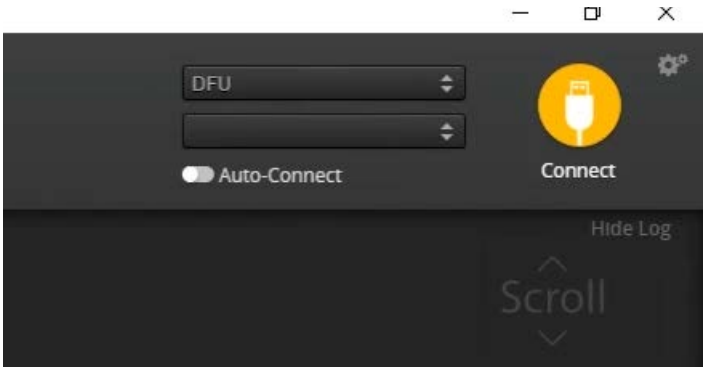
At this point, if you want to update your firmware, here is how to do it. But if you are annoyed and frustrated at this point and just want to go fly, please, go for it! You don't have to be running the absolute latest firmware to have a good time. Just forget about this nonsense!

Still here?

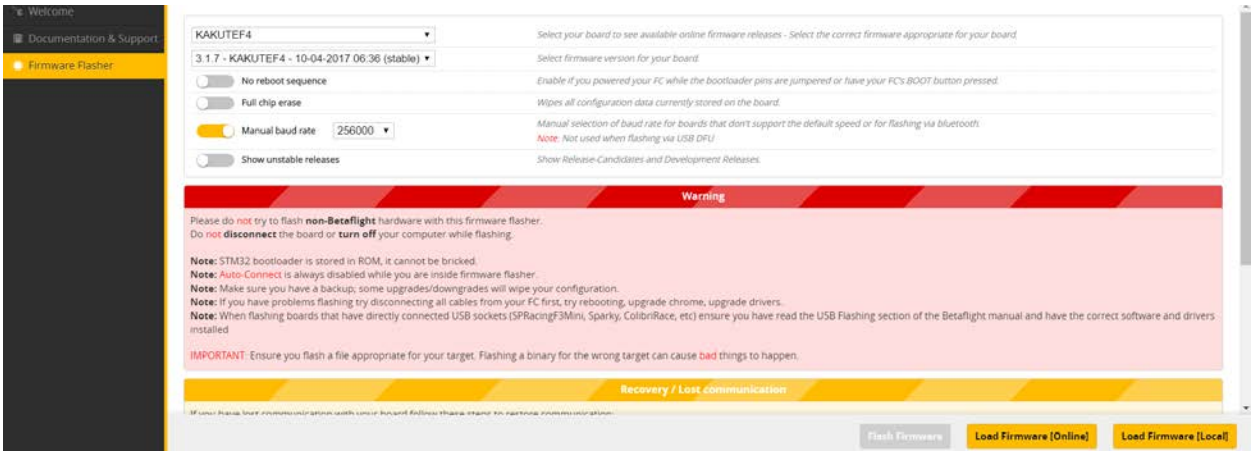
To flash your firmware, you must connect the board in "bootloader mode". Bootloader mode means that the board is ready to accept new programming. To put the board into bootloader mode, hold down the bootloader button while plugging in the USB cable. Leave the button pressed for a moment after plugging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

For the advanced users, a simpler way of getting into bootloader mode is to go to the CLI and type "bl". This will reboot the board into bootloader mode. On older versions of Betaflight, the command is "dfu" so try that if "bl" produces an error. This is especially useful if the board is installed in a copter where it isn't convenient to press the bootloader button.

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.

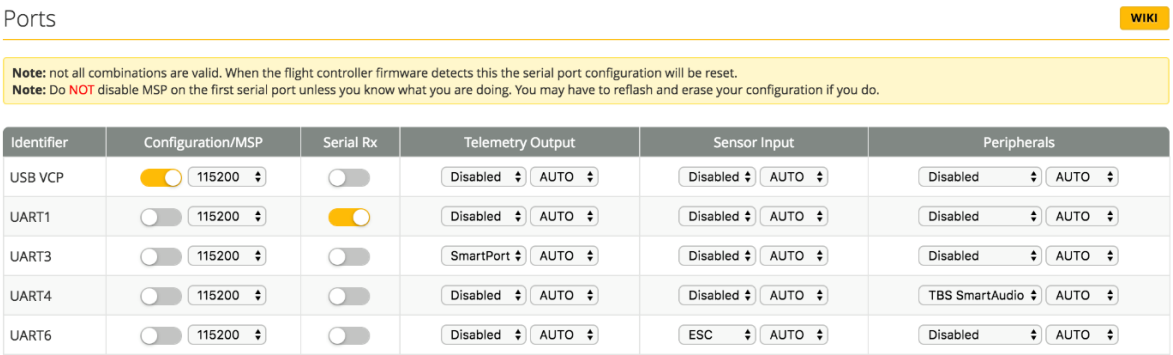
CONNECT TO THE BOARD

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. The instructions below assume that you used one of the recommended configurations from the Pre-Build Worksheet. If you've used a customized configuration, you can probably figure this part out for yourself.

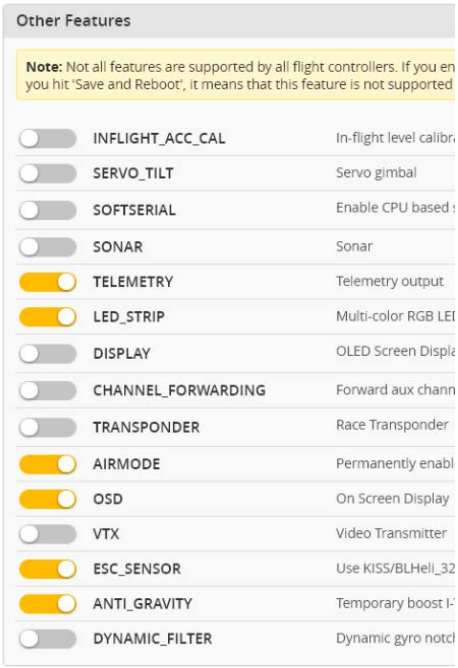


- USB VCP is the port that is used to talk between the JBF4 AIO and the Configurator GUI. MSP is the protocol that they use to talk to each other. If you ever turn off MSP on USB VCP, you will not be able to contact the board any more via the Configurator. You will have to re-flash the board and erase 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 vTX Remote Control, to control your video transmitter, on the UART4 line, in the Peripherals column, choose TBS SmartAudio from the pulldown menu.
- If you are using ESC Telemetry, in the Sensor Input column, choose ESC for whichever UART you used for the ESC Telemetry feature. If you prefer, you can also 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. However, we recommend that you leave these options at the default value and use the built-in sensors on the JBF4 AIO.

CONFIGURATION

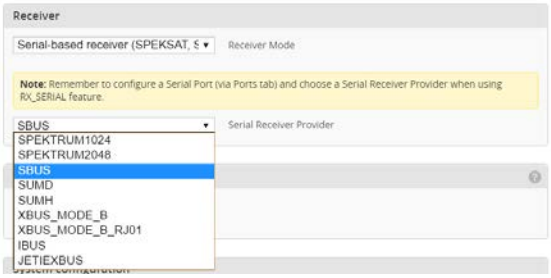
Click on the "Configuration" tab on the right side of the window. Scroll down to the "Other Features" section.

- If you are using SmartPort telemetry, enable TELEMETRY.
- 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 3.0 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. These are the most common serial receiver types that are likely to be encountered.



BLACKBOX

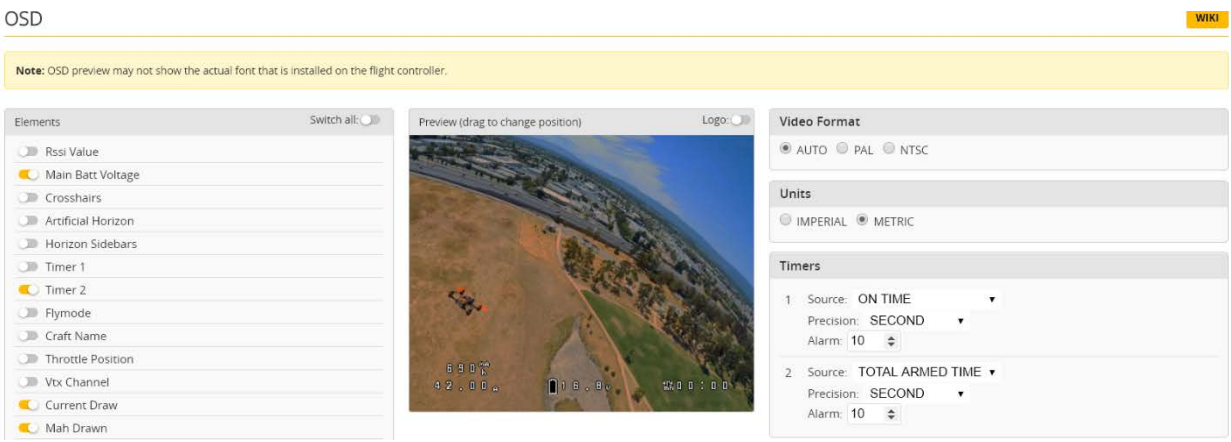
If you have enabled the Blackbox feature, go to the Blackbox tab on the left-hand side of the window. In the Blackbox tab, at the top, set the Blackbox Logging Device to SD Card and the Blackbox Logging Rate to 2 kHz.

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.



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.

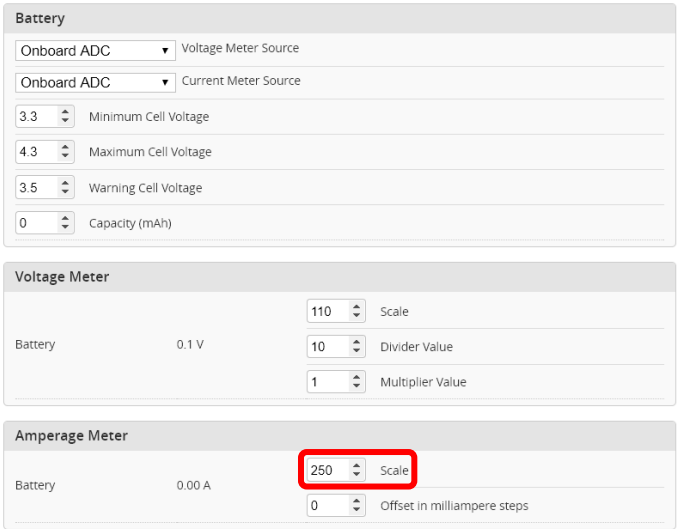
Although 3.3 volts is the recommended absolute minimum, you will get the best life out of your batteries if you keep the voltage above about 3.5 volts per cell—10.5 volts for 3S and 14.0 volts for 4S. If you are flying and you notice the battery going below this level, it might be time to land, or at 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. The current sensor on the board has been calibrated from the factory, but perfect calibration isn’t possible. We recommend that you record the mAh Drawn at the end of your first few flights, then compare it to the mAh that your battery charger puts back into the batteries. Then adjust the Current Scale in the Power And Battery tab to compensate for any difference. [Here is a video showing how to do that in more detail.]



Here is an example:

- OSD shows 1100 mAh Drawn at the end of the flight.
- Charger shows 1000 mAh put back in to the battery.
- 1100 / 1000 = 1.10. The OSD is reading 10% high.
- Current Scale works backwards from how you might think. To make the OSD read lower, you make Current Scale larger. Since the OSD is reading 10% high, we need to INCREASE Current Scale by 10%. This will cause the OSD to read 10% lower.
- If the Current Scale was at 400, we will multiply it by 1.10 to add 10%, meaning the new value should be 440.

The mAh put back in by the battery will seldom perfectly match the mAh reported by the OSD, but by taking several measurements and averaging the results, you can usually get it reasonably close.

SAVING YOUR CONFIGURATION

Once you have finished building, configuring, and tuning your multirotor, 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.

1. Connect your JBF4 AIO to your PC by plugging in USB.
2. Start the Betaflight GUI app.
3. Go to the CLI tab.
4. In the text box at the bottom of the CLI, type "diff" and hit enter. This will cause the flight controller to display all configuration options that you have changed from the default values.
5. Scroll back to the top of the window to find the line where you typed "diff".
6. Click your mouse and drag/scroll down to select all of the text below where you typed "diff". This is your configuration data.
7. Right-click and choose "Copy" to copy the highlighted text to the clipboard.
8. Open any text editor, such as Windows Notepad or MacOS TextEdit.
9. Right-click in the text editor window and choose "Paste" to paste the highlighted text into the editor.
10. Save the document somewhere you won't lose it.

To restore your configuration, do this:

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

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.